REPORT DOCUMENTATION PAGE

AFRL-SR-BL-TR-00-

0756

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	
	December, 1997	

4. TITLE AND SUBTITLE
1997 Summer Research Program (SRP), Summer Faculty Research Program (SFRP), Final Reports, Volume 4B, Rome Laboratory

5. FUNDING NUMBERS
F49620-93-C-0063

6. AUTHOR(S)
Gary Moore

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Research & Development Laboratories (RDL)
5800 Uplander Way
Culver City, CA 90230-6608

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Office of Scientific Research (AFOSR)
801 N. Randolph St.
Arlington, VA 22203-1977

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The United States Air Force Summer Research Program (USAF-SRP) is designed to introduce university, college, and technical institute faculty members, graduate students, and high school students to Air Force research. This is accomplished by the faculty members (Summer Faculty Research Program, (SFRP)), graduate students (Graduate Student Research Program (GSRP)), and high school students (High School Apprenticeship Program (HSAP)) being selected on a nationally advertised competitive basis during the summer intersession period to perform research at Air Force Research Laboratory (AFRL) Technical Directorates, Air Force Air Logistics Centers (ALC), and other AF Laboratories. This volume consists of a program overview, program management statistics, and the final technical reports from the SFRP participants at the Rome Laboratory.

14. SUBJECT TERMS
Air Force Research, Air Force, Engineering, Laboratories, Reports, Summer, Universities, Faculty, Graduate Student, High School Student

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

UNITED STATES AIR FORCE

SUMMER RESEARCH PROGRAM -- 1997

SUMMER FACULTY RESEARCH PROGRAM FINAL REPORTS

VOLUME 4B

ROME LABORATORY

RESEARCH & DEVELOPMENT LABORATORIES

5800 Uplander Way

Culver City, CA 90230-6608

Program Director, RDL
Gary Moore

Program Manager, AFOSR
Major Linda Steel-Goodwin

Program Manager, RDL
Scott Licoscos

Program Administrator, RDL
Johnetta Thompson

Program Administrator, RDL
Rebecca Kelly-Clemmons

Submitted to:

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH

Bolling Air Force Base

Washington, D.C.

December 1997

PREFACE

Reports in this volume are numbered consecutively beginning with number 1. Each report is paginated with the report number followed by consecutive page numbers, e.g., 1-1, 1-2, 1-3; 2-1, 2-2, 2-3.

Due to its length, Volume 4 is bound in two parts, 4A and 4B. Volume 4A contains #1-18. Volume 4B contains reports #19-31. The Table of Contents for Volume 4 is included in both parts.

This document is one of a set of 16 volumes describing the 1997 AFOSR Summer Research Program. The following volumes comprise the set:

VOLUME	TITLE
1	Program Management Report
	Summer Faculty Research Program (SFRP) Reports
2A & 2B	Armstrong Laboratory
3A & 3B	Phillips Laboratory
4A & 4B	Rome Laboratory
5A , 5B & 5C	Wright Laboratory
6	Arnold Engineering Development Center, United States Air Force Academy and Air Logistics Centers
	Graduate Student Research Program (GSRP) Reports
7A & 7B	Armstrong Laboratory
8	Phillips Laboratory
9	Rome Laboratory
10A & 10B	Wright Laboratory
11	Arnold Engineering Development Center, United States Air Force Academy, Wilford Hall Medical Center and Air Logistics Centers
	High School Apprenticeship Program (HSAP) Reports
12A & 12B	Armstrong Laboratory
13	Phillips Laboratory
14	Rome Laboratory
15A & 15B	Wright Laboratory
16	Arnold Engineering Development Center

SRP Final Report Table of Contents

SRP Final Report Table of Contents

SRP Final Report Table of Contents

SRP Final Report Table of Contents

SRP Final Report Table of Contents

SRP Final Report Table of Contents

SRP Final Report Table of Contents

SRP Final Report Table of Contents

1. INTRODUCTION

The Summer Research Program (SRP), sponsored by the Air Force Office of Scientific Research (AFOSR), offers paid opportunities for university faculty, graduate students, and high school students to conduct research in U.S. Air Force research laboratories nationwide during the summer.

Introduced by AFOSR in 1978, this innovative program is based on the concept of teaming academic researchers with Air Force scientists in the same disciplines using laboratory facilities and equipment not often available at associates' institutions.

The Summer Faculty Research Program (SFRP) is open annually to approximately 150 faculty members with at least two years of teaching and/or research experience in accredited U.S. colleges, universities, or technical institutions. SFRP associates must be either U.S. citizens or permanent residents.

The Graduate Student Research Program (GSRP) is open annually to approximately 100 graduate students holding a bachelor's or a master's degree; GSRP associates must be U.S. citizens enrolled full time at an accredited institution.

The High School Apprentice Program (HSAP) annually selects about 125 high school students located within a twenty mile commuting distance of participating Air Force laboratories.

AFOSR also offers its research associates an opportunity, under the Summer Research Extension Program (SREP), to continue their AFOSR-sponsored research at their home institutions through the award of research grants. In 1994 the maximum amount of each grant was increased from $20,000 to $25,000, and the number of AFOSR-sponsored grants decreased from 75 to 60. A separate annual report is compiled on the SREP.

The numbers of projected summer research participants in each of the three categories and SREP "grants" are usually increased through direct sponsorship by participating laboratories.

AFOSR's SRP has well served its objectives of building critical links between Air Force research laboratories and the academic community, opening avenues of communications and forging new research relationships between Air Force and academic technical experts in areas of national interest, and strengthening the nation's efforts to sustain careers in science and engineering. The success of the SRP can be gauged from its growth from inception (see Table 1) and from the favorable responses the 1997 participants expressed in end-of-tour SRP evaluations (Appendix B).

AFOSR contracts for administration of the SRP by civilian contractors. The contract was first awarded to Research & Development Laboratories (RDL) in September 1990. After completion of the

1990 contract, RDL (in 1993) won the recompetition for the basic year and four 1-year options.

2. PARTICIPATION IN THE SUMMER RESEARCH PROGRAM

The SRP began with faculty associates in 1979: graduate students were added in 1982 and high school students in 1986. The following table shows the number of associates in the program each year.

YEAR	SRP Participation, by Year			TOTAL
	SFRP	GSRP	HSAP	
1979	70			70
1980	87			87
1981	87			87
1982	91	17		108
1983	101	53		154
1984	152	84		236
1985	154	92		246
1986	158	100	42	300
1987	159	101	73	333
1988	153	107	101	361
1989	168	102	103	373
1990	165	121	132	418
1991	170	142	132	~~444~~
1992	185	121	159	~~464~~
1993	187	117	136	~~440~~
1994	192	117	133	~~442~~
1995	190	115	137	~~442~~
1996	188	109	138	435
1997	148	98	140	427

Beginning in 1993, due to budget cuts, some of the laboratories weren't able to afford to fund as many associates as in previous years. Since then, the number of funded positions has remained fairly constant at a slightly lower level.

3. RECRUITING AND SELECTION

The SRP is conducted on a nationally advertised and competitive-selection basis. The advertising for faculty and graduate students consisted primarily of the mailing of 8,000 52-page SRP brochures to chairpersons of departments relevant to AFOSR research and to administrators of grants in accredited universities, colleges, and technical institutions. Historically Black Colleges and Universities (HBCUs) and Minority Institutions (MIs) were included. Brochures also went to all participating USAF laboratories, the previous year's participants, and numerous individual requesters (over 1000 annually).

RDL placed advertisements in the following publications: *Black Issues in Higher Education, Winds of Change,* and *IEEE Spectrum.* Because no participants list either *Physics Today* or *Chemical & Engineering News* as being their source of learning about the program for the past several years, advertisements in these magazines were dropped, and the funds were used to cover increases in brochure printing costs.

High school applicants can participate only in laboratories located no more than 20 miles from their residence. Tailored brochures on the HSAP were sent to the head counselors of 180 high schools in the vicinity of participating laboratories, with instructions for publicizing the program in their schools. High school students selected to serve at Wright Laboratory's Armament Directorate (Eglin Air Force Base, Florida) serve eleven weeks as opposed to the eight weeks normally worked by high school students at all other participating laboratories.

Each SFRP or GSRP applicant is given a first, second, and third choice of laboratory. High school students who have more than one laboratory or directorate near their homes are also given first, second, and third choices.

Laboratories make their selections and prioritize their nominees. AFOSR then determines the number to be funded at each laboratory and approves laboratories' selections.

Subsequently, laboratories use their own funds to sponsor additional candidates. Some selectees do not accept the appointment, so alternate candidates are chosen. This multi-step selection procedure results in some candidates being notified of their acceptance after scheduled deadlines. The total applicants and participants for 1997 are shown in this table.

1997 Applicants and Participants			
PARTICIPANT CATEGORY	TOTAL APPLICANTS	SELECTEES	DECLINING SELECTEES
SFRP	490	188	32
(HBCU/MI)	(0)	(0)	(0)
GSRP	202	98	9
(HBCU/MI)	(0)	(0)	(0)
HSAP	433	140	14
TOTAL	1125	426	55

4. SITE VISITS

During June and July of 1997, representatives of both AFOSR/NI and RDL visited each participating laboratory to provide briefings, answer questions, and resolve problems for both laboratory personnel and participants. The objective was to ensure that the SRP would be as constructive as possible for all participants. Both SRP participants and RDL representatives found these visits beneficial. At many of the laboratories, this was the only opportunity for all participants to meet at one time to share their experiences and exchange ideas.

5. HISTORICALLY BLACK COLLEGES AND UNIVERSITIES AND MINORITY INSTITUTIONS (HBCU/MIs)

Before 1993, an RDL program representative visited from seven to ten different HBCU/MIs annually to promote interest in the SRP among the faculty and graduate students. These efforts were marginally effective, yielding a doubling of HBCI/MI applicants. In an effort to achieve AFOSR's goal of 10% of all applicants and selectees being HBCU/MI qualified, the RDL team decided to try other avenues of approach to increase the number of qualified applicants. Through the combined efforts of the AFOSR Program Office at Bolling AFB and RDL, two very active minority groups were found, HACU (Hispanic American Colleges and Universities) and AISES (American Indian Science and Engineering Society). RDL is in communication with representatives of each of these organizations on a monthly basis to keep up with the their activities and special events. Both organizations have widely-distributed magazines/quarterlies in which RDL placed ads.

Since 1994 the number of both SFRP and GSRP HBCU/MI applicants and participants has increased ten-fold, from about two dozen SFRP applicants and a half dozen selectees to over 100 applicants and two dozen selectees, and a half-dozen GSRP applicants and two or three selectees to 18 applicants and 7 or 8 selectees. Since 1993, the SFRP had a two-fold applicant increase and a two-fold selectee increase. Since 1993, the GSRP had a three-fold applicant increase and a three to four-fold increase in selectees.

In addition to RDL's special recruiting efforts, AFOSR attempts each year to obtain additional funding or use leftover funding from cancellations the past year to fund HBCU/MI associates. This year, 5 HBCU/MI SFRPs declined after they were selected (and there was no one qualified to replace them with). The following table records HBCU/MI participation in this program.

SRP HBCU/MI Participation, By Year				
YEAR	SFRP		GSRP	
	Applicants	Participants	Applicants	Participants
1985	76	23	15	11
1986	70	18	20	10
1987	82	32	32	10
1988	53	17	23	14
1989	39	15	13	4
1990	43	14	17	3
1991	42	13	8	5
1992	70	13	9	5
1993	60	13	6	2
1994	90	16	11	6
1995	90	21	20	8
1996	119	27	18	7

6. SRP FUNDING SOURCES

Funding sources for the 1997 SRP were the AFOSR-provided slots for the basic contract and laboratory funds. Funding sources by category for the 1997 SRP selected participants are shown here.

1997 SRP FUNDING CATEGORY	SFRP	GSRP	HSAP
AFOSR Basic Allocation Funds	141	89	123
USAF Laboratory Funds	48	9	17
HBCU/MI By AFOSR (Using Procured Addn'l Funds)	0	0	N/A
TOTAL	9	98	140

SFRP - 188 were selected, but thirty two canceled too late to be replaced.
GSRP - 98 were selected, but nine canceled too late to be replaced.
HSAP - 140 were selected, but fourteen canceled too late to be replaced.

7. COMPENSATION FOR PARTICIPANTS

Compensation for SRP participants, per five-day work week, is shown in this table.

1997 SRP Associate Compensation

PARTICIPANT CATEGORY	1991	1992	1993	1994	1995	1996	1997
Faculty Members	$690	$718	$740	$740	$740	$770	$770
Graduate Student (Master's Degree)	$425	$442	$455	$455	$455	$470	$470
Graduate Student (Bachelor's Degree)	$365	$380	$391	$391	$391	$400	$400
High School Student (First Year)	$200	$200	$200	$200	$200	$200	$200
High School Student (Subsequent Years)	$240	$240	$240	$240	$240	$240	$240

The program also offered associates whose homes were more than 50 miles from the laboratory an expense allowance (seven days per week) of $50/day for faculty and $40/day for graduate students. Transportation to the laboratory at the beginning of their tour and back to their home destinations at the end was also reimbursed for these participants. Of the combined SFRP and GSRP associates, 65 % (194 out of 286) claimed travel reimbursements at an average round-trip cost of $776.

Faculty members were encouraged to visit their laboratories before their summer tour began. All costs of these orientation visits were reimbursed. Forty-three percent (85 out of 188) of faculty associates took orientation trips at an average cost of $388. By contrast, in 1993, 58 % of SFRP associates took

orientation visits at an average cost of $685; that was the highest percentage of associates opting to take an orientation trip since RDL has administered the SRP, and the highest average cost of an orientation trip. These 1993 numbers are included to show the fluctuation which can occur in these numbers for planning purposes.

Program participants submitted biweekly vouchers countersigned by their laboratory research focal point, and RDL issued paychecks so as to arrive in associates' hands two weeks later.

This is the second year of using direct deposit for the SFRP and GSRP associates. The process went much more smoothly with respect to obtaining required information from the associates, only 7% of the associates' information needed clarification in order for direct deposit to properly function as opposed to 10% from last year. The remaining associates received their stipend and expense payments via checks sent in the US mail.

HSAP program participants were considered actual RDL employees, and their respective state and federal income tax and Social Security were withheld from their paychecks. By the nature of their independent research, SFRP and GSRP program participants were considered to be consultants or independent contractors. As such, SFRP and GSRP associates were responsible for their own income taxes, Social Security, and insurance.

8. CONTENTS OF THE 1997 REPORT

The complete set of reports for the 1997 SRP includes this program management report (Volume 1) augmented by fifteen volumes of final research reports by the 1997 associates, as indicated below:

1997 SRP Final Report Volume Assignments

LABORATORY	SFRP	GSRP	HSAP
Armstrong	2	7	12
Phillips	3	8	13
Rome	4	9	14
Wright	5A, 5B	10	15
AEDC, ALCs, WHMC	6	11	16

APPENDIX A -- PROGRAM STATISTICAL SUMMARY

A. Colleges/Universities Represented

Selected SFRP associates represented 169 different colleges, universities, and institutions, GSRP associates represented 95 different colleges, universities, and institutions.

B. States Represented

SFRP -Applicants came from 47 states plus Washington D.C. Selectees represent 44 states.

GSRP - Applicants came from 44 states. Selectees represent 32 states.

HSAP - Applicants came from thirteen states. Selectees represent nine states.

Total Number of Participants	
SFRP	189
GSRP	97
HSAP	140
TOTAL	426

Degrees Represented			
	SFRP	GSRP	TOTAL
Doctoral	184	0	184
Master's	2	41	43
Bachelor's	0	56	56
TOTAL	186	97	298

SFRP Academic Titles	
Assistant Professor	64
Associate Professor	70
Professor	40
Instructor	0
Chairman	1
Visiting Professor	1
Visiting Assoc. Prof.	1
Research Associate	9
TOTAL	186

Source of Learning About the SRP		
Category	Applicants	Selectees
Applied/participated in prior years	28%	34%
Colleague familiar with SRP	19%	16%
Brochure mailed to institution	23%	17%
Contact with Air Force laboratory	17%	23%
IEEE Spectrum	2%	1%
BIIHE	1%	1%
Other source	10%	8%
TOTAL	100%	100%

APPENDIX B – SRP EVALUATION RESPONSES

1. OVERVIEW

Evaluations were completed and returned to RDL by four groups at the completion of the SRP. The number of respondents in each group is shown below.

Table B-1. Total SRP Evaluations Received

Evaluation Group	Responses
SFRP & GSRPs	275
HSAPs	113
USAF Laboratory Focal Points	84
USAF Laboratory HSAP Mentors	6

All groups indicate unanimous enthusiasm for the SRP experience.

The summarized recommendations for program improvement from both associates and laboratory personnel are listed below:

A. Better preparation on the labs' part prior to associates' arrival (i.e., office space, computer assets, clearly defined scope of work).

B. Faculty Associates suggest higher stipends for SFRP associates.

C. Both HSAP Air Force laboratory mentors and associates would like the summer tour extended from the current 8 weeks to either 10 or 11 weeks; the groups state it takes 4-6 weeks just to get high school students up-to-speed on what's going on at laboratory. (Note: this same argument was used to raise the faculty and graduate student participation time a few years ago.)

2. 1997 USAF LABORATORY FOCAL POINT (LFP) EVALUATION RESPONSES

The summarized results listed below are from the 84 LFP evaluations received.

1. LFP evaluations received and associate preferences:

Table B-2. Air Force LFP Evaluation Responses (By Type)

		How Many Associates Would You Prefer To Get ? (% Response)											
		SFRP				GSRP (w/Univ Professor)				GSRP (w/o Univ Professor)			
Lab	Evals Recv'd	0	1	2	3+	0	1	2	3+	0	1	2	3+
AEDC	0	-	-	-	-	-	-	-	-	-	-	-	-
WHMC	0	-	-	-	-	-	-	-	-	-	-	-	-
AL	7	28	28	28	14	54	14	28	0	86	0	14	0
USAFA	1	0	100	0	0	100	0	0	0	0	100	0	0
PL	25	40	40	16	4	88	12	0	0	84	12	4	0
RL	5	60	40	0	0	80	10	0	0	100	0	0	0
WL	46	30	43	20	6	78	17	4	0	93	4	2	0
Total	84	32%	50%	13%	5%	80%	11%	6%	0%	73%	23%	4%	0%

LFP Evaluation Summary. The summarized responses, by laboratory, are listed on the following page. LFPs were asked to rate the following questions on a scale from 1 (below average) to 5 (above average).

2. LFPs involved in SRP associate application evaluation process:
   a. Time available for evaluation of applications:
   b. Adequacy of applications for selection process:
3. Value of orientation trips:
4. Length of research tour:
5. a. Benefits of associate's work to laboratory:
   b. Benefits of associate's work to Air Force:
6. a. Enhancement of research qualifications for LFP and staff:
   b. Enhancement of research qualifications for SFRP associate:
   c. Enhancement of research qualifications for GSRP associate:
7. a. Enhancement of knowledge for LFP and staff:
   b. Enhancement of knowledge for SFRP associate:
   c. Enhancement of knowledge for GSRP associate:
8. Value of Air Force and university links:
9. Potential for future collaboration:
10. a. Your working relationship with SFRP:
    b. Your working relationship with GSRP:
11. Expenditure of your time worthwhile:

(Continued on next page)

12. Quality of program literature for associate:
13. a. Quality of RDL's communications with you:
    b. Quality of RDL's communications with associates:
14. Overall assessment of SRP:

Table B-3. Laboratory Focal Point Reponses to above questions

	AEDC	*AL*	*USAFA*	*PL*	*RL*	*WHMC*	*WL*
# Evals Recv'd	0	7	1	14	5	0	46
Question #							
2	-	86 %	0 %	88 %	80 %	-	85 %
2a	-	4.3	n/a	3.8	4.0	-	3.6
2b	-	4.0	n/a	3.9	4.5	-	4.1
3	-	4.5	n/a	4.3	4.3	-	3.7
4	-	4.1	4.0	4.1	4.2	-	3.9
5a	-	4.3	5.0	4.3	4.6	-	4.4
5b	-	4.5	n/a	4.2	4.6	-	4.3
6a	-	4.5	5.0	4.0	4.4	-	4.3
6b	-	4.3	n/a	4.1	5.0	-	4.4
6c	-	3.7	5.0	3.5	5.0	-	4.3
7a	-	4.7	5.0	4.0	4.4	-	4.3
7b	-	4.3	n/a	4.2	5.0	-	4.4
7c	-	4.0	5.0	3.9	5.0	-	4.3
8	-	4.6	4.0	4.5	4.6	-	4.3
9	-	4.9	5.0	4.4	4.8	-	4.2
10a	-	5.0	n/a	4.6	4.6	-	4.6
10b	-	4.7	5.0	3.9	5.0	-	4.4
11	-	4.6	5.0	4.4	4.8	-	4.4
12	-	4.0	4.0	4.0	4.2	-	3.8
13a	-	3.2	4.0	3.5	3.8	-	3.4
13b	-	3.4	4.0	3.6	4.5	-	3.6
14	-	4.4	5.0	4.4	4.8	-	4.4

3. 1997 SFRP & GSRP EVALUATION RESPONSES

The summarized results listed below are from the 257 SFRP/GSRP evaluations received.

Associates were asked to rate the following questions on a scale from 1 (below average) to 5 (above average) - by Air Force base results and over-all results of the 1997 evaluations are listed after the questions.

1. The match between the laboratories research and your field:
2. Your working relationship with your LFP:
3. Enhancement of your academic qualifications:
4. Enhancement of your research qualifications:
5. Lab readiness for you: LFP, task, plan:
6. Lab readiness for you: equipment, supplies, facilities:
7. Lab resources:
8. Lab research and administrative support:
9. Adequacy of brochure and associate handbook:
10. RDL communications with you:
11. Overall payment procedures:
12. Overall assessment of the SRP:
13. a. Would you apply again?
    b. Will you continue this or related research?
14. Was length of your tour satisfactory?
15. Percentage of associates who experienced difficulties in finding housing:
16. Where did you stay during your SRP tour?
    a. At Home:
    b. With Friend:
    c. On Local Economy:
    d. Base Quarters:
17. Value of orientation visit:
    a. Essential:
    b. Convenient:
    c. Not Worth Cost:
    d. Not Used:

SFRP and GSRP associate's responses are listed in tabular format on the following page.

Table B-4. 1997 SFRP & GSRP Associate Responses to SRP Evaluation

	Arnold	Brooks	Edwards	Eglin	Griffis	Hanscom	Kelly	Kirtland	Lackland	Robins	Tyndall	WPAFB	average
# res	6	48	6	14	31	19	3	32	1	2	10	85	257
1	4.8	4.4	4.6	4.7	4.4	4.9	4.6	4.6	5.0	5.0	4.0	4.7	4.6
2	5.0	4.6	4.1	4.9	4.7	4.7	5.0	4.7	5.0	5.0	4.6	4.8	4.7
3	4.5	4.4	4.0	4.6	4.3	4.2	4.3	4.4	5.0	5.0	4.5	4.3	4.4
4	4.3	4.5	3.8	4.6	4.4	4.4	4.3	4.6	5.0	4.0	4.4	4.5	4.5
5	4.5	4.3	3.3	4.8	4.4	4.5	4.3	4.2	5.0	5.0	3.9	4.4	4.4
6	4.3	4.3	3.7	4.7	4.4	4.5	4.0	3.8	5.0	5.0	3.8	4.2	4.2
7	4.5	4.4	4.2	4.8	4.5	4.3	4.3	4.1	5.0	5.0	4.3	4.3	4.4
8	4.5	4.6	3.0	4.9	4.4	4.3	4.3	4.5	5.0	5.0	4.7	4.5	4.5
9	4.7	4.5	4.7	4.5	4.3	4.5	4.7	4.3	5.0	5.0	4.1	4.5	4.5
10	4.2	4.4	4.7	4.4	4.1	4.1	4.0	4.2	5.0	4.5	3.6	4.4	4.3
11	3.8	4.1	4.5	4.0	3.9	4.1	4.0	4.0	3.0	4.0	3.7	4.0	4.0
12	5.7	4.7	4.3	4.9	4.5	4.9	4.7	4.6	5.0	4.5	4.6	4.5	4.6
Numbers below are percentages													
13a	83	90	83	93	87	75	100	81	100	100	100	86	87
13b	100	89	83	100	94	98	100	94	100	100	100	94	93
14	83	96	100	90	87	80	100	92	100	100	70	84	88
15	17	6	0	33	20	76	33	25	0	100	20	8	39
16a	-	26	17	9	38	23	33	4	-	-	-	30	
16b	100	33	-	40	-	8	-	-	-	-	36	2	
16c	-	41	83	40	62	69	67	96	100	100	64	68	
16d	-	-	-	-	-	-	-	-	-	-	-	0	
17a	-	33	100	17	50	14	67	39	-	50	40	31	35
17b	-	21	-	17	10	14	-	24	-	50	20	16	16
17c	-	-	-	-	10	7	-	-	-	-	-	2	3
17d	100	46	-	66	30	69	33	37	100	-	40	51	46

4. 1997 USAF LABORATORY HSAP MENTOR EVALUATION RESPONSES

Not enough evaluations received (5 total) from Mentors to do useful summary.

5. 1997 HSAP EVALUATION RESPONSES

The summarized results listed below are from the 113 HSAP evaluations received.

HSAP apprentices were asked to rate the following questions on a scale from 1 (below average) to 5 (above average)

1. Your influence on selection of topic/type of work.
2. Working relationship with mentor, other lab scientists.
3. Enhancement of your academic qualifications.
4. Technically challenging work.
5. Lab readiness for you: mentor, task, work plan, equipment.
6. Influence on your career.
7. Increased interest in math/science.
8. Lab research & administrative support.
9. Adequacy of RDL's Apprentice Handbook and administrative materials.
10. Responsiveness of RDL communications.
11. Overall payment procedures.
12. Overall assessment of SRP value to you.
13. Would you apply again next year? Yes (92 %)
14. Will you pursue future studies related to this research? Yes (68 %)
15. Was Tour length satisfactory? Yes (82 %)

	Arnold	Brooks	Edwards	Eglin	Griffiss	Hanscom	Kirtland	Tyndall	WPAFB	Totals
# resp	5	19	7	15	13	2	7	5	40	113
1	2.8	3.3	3.4	3.5	3.4	4.0	3.2	3.6	3.6	3.4
2	4.4	4.6	4.5	4.8	4.6	4.0	4.4	4.0	4.6	4.6
3	4.0	4.2	4.1	4.3	4.5	5.0	4.3	4.6	4.4	4.4
4	3.6	3.9	4.0	4.5	4.2	5.0	4.6	3.8	4.3	4.2
5	4.4	4.1	3.7	4.5	4.1	3.0	3.9	3.6	3.9	4.0
6	3.2	3.6	3.6	4.1	3.8	5.0	3.3	3.8	3.6	3.7
7	2.8	4.1	4.0	3.9	3.9	5.0	3.6	4.0	4.0	3.9
8	3.8	4.1	4.0	4.3	4.0	4.0	4.3	3.8	4.3	4.2
9	4.4	3.6	4.1	4.1	3.5	4.0	3.9	4.0	3.7	3.8
10	4.0	3.8	4.1	3.7	4.1	4.0	3.9	2.4	3.8	3.8
11	4.2	4.2	3.7	3.9	3.8	3.0	3.7	2.6	3.7	3.8
12	4.0	4.5	4.9	4.6	4.6	5.0	4.6	4.2	4.3	4.5
Numbers below are percentages										
13	60%	95%	100%	100%	85%	100%	100%	100%	90%	92%
14	20%	80%	71%	80%	54%	100%	71%	80%	65%	68%
15	100%	70%	71%	100%	100%	50%	86%	60%	80%	82%

GAIN SPECTRA OF BEAM-COUPLING IN PHOTOREFRACTIVE SEMICONDUCTORS

Jeffrey B. Norman
Associate Professor
Department of Physics and Astronomy

Vassar College
Box 559
Poughkeepsie, NY 12604

Final Report for:
Summer Research Program
Rome Laboratory

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, Washington, DC

And

Rome Laboratory

August 1997

GAIN SPECTRA OF BEAM-COUPLING IN PHOTOREFRACTIVE SEMICONDUCTORS

Jeffrey B. Norman
Associate Professor of Physics
Department of Physics and Astronomy
Vassar College
Box 559
Poughkeepsie, NY 12604

G. Brost
Photonics Center, Rome Laboratory

S. Odoulov, K. Shcherbin, A. Shumelyuk and V. Taranov.
Department of Quantum Electronics
Institute of Physics of the National Academy of Sciences of Ukraine

Abstract

Photorefractive recording dynamics of two-beam coupling in semi-insulating semiconductors by beams with slightly different frequencies is studied theoretically and experimentally. The influence of bulk absorption, Gaussian beam profiles, and experimental geometry on the temporal response are analyzed. These effects act to narrow the bandwidth. Measurement of the material photorefractive time constant is discussed.

GAIN SPECTRA OF BEAM-COUPLING IN PHOTOREFRACTIVE SEMICONDUCTORS

Jeffrey B. Norman, G. Brost, S. Odoulov, K. Shcherbin, A. Shumelyuk, and V. Taranov.

1. Introduction

Measurement of the temporal response of two-beam coupling in photorefractive materials is important for determination of material parameters and for device characterization. However, there are a number of experimental and material factors that can strongly influence such measurements. These factors include bulk absorption, beam intensity profiles, coupling geometry, coupling strength, modulation depth, dark current, and pump depletion. Even under conditions in which only absorption appears to be a factor, the bandwidth of the material can be drastically different than that predicted by the standard photorefractive theory.

In fast photorefractive materials, such as the II-VI and III-V semiconductors, it is often convenient to measure the temporal response in the frequency domain rather than the time domain. This approach avoids the requirement of fast shutters and detectors, and has the advantage that measurements in the frequency domain are carried out entirely in the steady state regime. In addition, for purposes of accounting for the influences on the temporal response mentioned above, work in the frequency domain is particularly convenient since modelling is often more readily done there.

In the diffusion regime, the standard solution of the material equations for a material with one kind of photorefractive trap predicts that the time dependence of the gain coefficient is given by[1]

$$\Gamma = \Gamma_0\left(1 - e^{-t/\tau}\right) \quad , \tag{1}$$

where Γ_0 is the steady state gain coefficient and τ is the photorefractive time constant. In the frequency domain, the two-beam coupling gain spectra for moving gratings is given by[2]

$$\Gamma = \Gamma_0 / \left(1 - \Omega^2\tau^2\right) \quad , \tag{2}$$

where Ω is the angular frequency detuning between the pump and signal beams. Eq. (2) is a simple Lorentzian function centered at Ω = 0 Hz. Derivation of Eqs. (1) and (2) assumes negligible contributions from the factors mentioned in the first paragraph. In particular, it assumes a lossless material and plane wave illumination. In practice, these conditions cannot be met. The photorefractive response time is dependent on intensity and is therefore position dependent in the material.

In this paper, we study the influence of a spatially dependent light intensity distribution on the gain spectra of the photorefractive. It is assumed here that dark current, large modulation, and pump depletion are not present, and that the gain is in the small signal regime. We present a theoretical analysis which elucidates the effects of optical absorption and Gaussian beam profiles. We also present experimental results of gain spectra and temporal response measured in photorefractive semiconductors. It is found that the frequency response can significantly depart from the Lorentzian shape of Eq. (2). We use a simple method of eliminating these influences for the purpose of measuring the material photorefractive time constant.

2. Theory

2.1 Absorption

In this section, we consider the effect of optical absorption on the gain spectra. We assume that the pump beam provides uniform illumination distribution in the transverse direction but falls off exponentially with distance according to Beer's Law, $I(z) = I_0\exp(-\alpha z)$. We also assume that the photorefractive response time is inversely proportional to some power q of the light intensity, so that the position dependent response time is given by

$$\tau(z)=\tau_0(I_0/I(z))^q=\tau_0 e^{q\alpha z}, \tag{3}$$

where τ_0 is the photorefractive time constant at intensity I_0, at the front of the crystal. The integral gain coefficient for moving gratings, assuming symmetric geometry and a large pump to signal intensity ratio, is then given by

$$\Gamma = \frac{1}{L}\int_0^L \frac{\Gamma_0}{1+\Omega^2\tau^2(z)}dz, \tag{4}$$

where L is the interaction length. After substitution for $\tau(z)$ and integration, the final expression for the gain spectrum becomes

$$\Gamma = \Gamma_0\left[1+\frac{1}{2q\alpha L}\ln\left(\frac{1+\Omega^2\tau_0^2}{1+\Omega^2\tau_0^2 e^{2q\alpha L}}\right)\right]. \tag{5}$$

This result agrees with the transfer function derived by Hermanns *et al*[3] for $q = 1$. Fig. 1 demonstrates the effect of absorption on the gain spectra. Here we plot the gain coefficient as a function of the normalized frequency detuning for three different values of αL with $q = 1$; (a) α=0, (b) $\alpha L = 0.72$, and (c) $\alpha L = 5$. Curve *a* is the single Lorentzian profile given by Eq. (2) for no absorption. Curve *b* reflects an αL for typical crystal parameters. In this case, the gain profile can still be characterized as single Lorentzian, but narrower than for no absorption. This result shows

that the effect of absorption for most crystals is to increase the exponential time constant. In this case, by about 50%. Curve *c* demonstrates the effect of very large absorption. In this case, the gain profile is much narrower (by about a factor of ten) than the no absorption case. It can no longer be characterized as a single Lorentzian. In fact, it is more accurately described as the sum of two Lorentzians.

2.2 Gaussian beam profiles

We now consider the more general case of finite beam sizes. The geometry is as shown in Fig. 2. The signal beam I_S and pump beam I_P cross inside the crystal of thickness *d* at an angle of 2θ. The intensity profiles of the signal and pump beams are assumed to be Gaussian. As before, we assume that the pump beam intensity is much larger than the signal beam intensity, so that $\tau(I)$ is determined by I_P. The gain coefficient is calculated by integrating the differential gain coefficient over the volume of the signal beam. After a coordinate transformation from the primed to the unprimed coordinates, and taking the origin to be where the beams cross, the gain coefficient is given by

$$\Gamma = \frac{\Gamma_0}{\pi\omega_S^2 d/2\cos(\theta)} \int_{\frac{\xi d}{\cos(\theta)}}^{\frac{(1-\xi)d}{\cos(\theta)}} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{e^{\frac{2(x^2+y^2)}{\omega_S^2}}\, dydxdz}{1+\Omega^2\tau_0^2 e^{2q\alpha\left[-\sin(2\theta)x+\cos(2\theta)z+\frac{\xi d}{\cos(\theta)}\right]} e^{\frac{4[\cos(2\theta)x+\sin(2\theta)z]^2+4y^2}{\omega_P^2}}}, \quad (6)$$

where $2\omega_S$ and $2\omega_P$ are the $1/e^2$ intensity diameters of the signal and pump beams, respectively. Here, the differential gain was weighted according to the Gaussian profile of the signal beam, *d* is the crystal thickness, and ξ is a parameter that indicates the location of the origin. The integral in Eq. (8) was evaluated numerically.

There are seven parameters that influence the gain coefficient in Eq. (6); *q*, α, *d*, ω_S, ω_P, q, and ξ. It is not possible here to fully evaluate the effect of each parameter. We highlight some of the more important features. In the following analysis, we let $q = 1$, $\alpha = 1.2$ cm^{-1}, $d = 0.6$ cm, and $2\omega_S = 2$ mm.

In Fig. 3, the calculated gain spectra are plotted for crossing angles of $2\theta = 0°$ and 25°, with $2\omega_P = 2$ mm and the beams crossing in the center of the crystal ($\xi = 1/2$). The gain calculated from Eq. (5) for plane waves is also shown for comparison (curve a). The position dependent intensity associated with the Gaussian beam profile causes a significant narrowing of the

bandwidth as compared to that of the plane wave case: a factor of about five for $2\theta = 25°$. The resulting gain spectrum is similar to that of the plane wave large absorption example in Fig. 1.

The influence of the Gaussian beam profile can be minimized with an expanded pump beam. Fig. 4 shows the calculated gain for a crossing angle of $2\theta = 25°$, $\xi = 1/2$, and $\omega_P = \omega_S$, $2\omega_S$, and $5\omega_S$. There is still appreciable narrowing of the gain spectrum for $\omega_P = 2\omega_S$, but the spectrum is not much different than the plane wave case for $\omega_P = 5\omega_S$.

In the previous examples, the beams were assumed to cross at the center of the crystal. Fig. 5 shows the calculated gain spectra for different crossing locations, $\xi = 0$, 1/2, and 1, corresponding to the front, center, and back of the crystal, with $\omega_P = 3\omega_S$ and a crossing angle of $2\theta = 25°$. Although the steady state gain in the degenerate case does not depend on ξ, the bandwidth does. The widest bandwidth is obtained when the beams cross at the center of the crystal.

2.3 Discussion

We comment here on some limits of applicability in using Eq. (6). The main limitation concerns the assumption that the pump beam intensity is significantly greater than that of the signal beam. This applies not just to the peak intensities, but throughout the volume of the signal beam. Equation (6) assumes that the time dependence is determined by the pump beam intensity and not by that of the signal beam. Also, this approach does not account for the effects of large modulation, beam depletion, or dark current, which could occur in the Gaussian wings. Consequently, the limit of application depends upon the geometry of the problem. The examples presented here are valid for beam ratios greater than 100. In the case of small beam diameters and thick crystals, the photorefractive interaction may occur far enough out in the wings of the pump beam that these assumptions are no longer valid. For these problems, the numerical approach of Fluck *et al.* [4] is more appropriate.

The analysis did not consider the effects of beam coupling on the time constant. Analysis of the photorefractive response that includes the interaction of the optical field with the space charge field has shown a decreased bandwidth for materials with large ΓL, such as the ferroelectrics.[5,6] For semiconductors, the gain is sufficiently small that this effect is negligible.

A non-Lorentzian gain spectrum may be obtained under certain experimental conditions, even in the plane wave, no absorption case. The nonlinearity of the photorefractive response is such that at large modulation the response becomes superlinear, but with a very slow temporal

component.[7] A calculation of the gain spectra for different values of m is shown in Fig. 6. These results are solutions of the material equations and reflect the local response of plane waves. For $m < 0.6$ the response does not deviate much from the small m result of Eq. (2). For larger values of m, the space charge field is enhanced, but only for small frequency shifts, due to the slow time component.

3. Experimental results

In this section, we present measured two-beam coupling frequency and time response data. As predicted in Section 2, it is found that the temporal response varies considerably with changes in Gaussian beam sizes and coupling geometry, and is influenced strongly by the bulk absorption of the material.

The experimental configuration for our measurements of two-beam coupling frequency and time response is shown in Fig. 7. The moving grating was produced in a sample of GaAs:Cr by a linear phase modulation of one of the beams using an electro-optic phase modulator. In the frequency response measurements, the steady-state gain was measured as a function of the frequency, $f = \Omega / 2\pi$, of the ramp applied to the e-o modulator.

The two-beam coupling measurements were performed at a wavelength of 1.06μm and a grating period of 0.7μm. The laser beams were incident on the $(1\bar{1}0)$ face of the GaAs:Cr crystal, the grating vector was oriented in the $\langle 001 \rangle$ direction, and the beams were s-polarized. The dimensions of the crystal, in the directions $\langle 110 \rangle \times \langle 100 \rangle \times \langle 1\bar{1}0 \rangle$, were 11x10x6.1 mm^3 and the measured absorption coefficient was 1.2 cm^{-1}.

Our two-beam coupling frequency and time response data for the GaAs:Cr sample, corresponding to various illumination conditions, are shown in Figs. 8-12. In what follows, we will describe the effects of these illumination conditions and of the material absorption on the temporal response of our sample.

In Fig. 8, the Gaussian signal and pump beams have $1/e^2$ diameters of 1.8mm and 2.0mm, respectively. The data of Figs. 8(a) and 8(b), corresponding to the frequency and time response, respectively, were acquired under identical experimental conditions. Note that the beam diameters in this case are much smaller than the crystal thickness of 6.1mm. The dashed curve in Fig. 8(a) is an attempt to fit the data to the Lorentzian of Eq. (2) and it is apparent that the data depart significantly from this model. If the assumptions inherent in Eq. (2) were valid, the material time constant would be related to the full width at half maximum, Δ, of the Lorentzian by $\tau = 1/\pi\Delta$.

The discrepancies between the data in Fig. 8 and the response expected from the simple theory can be attributed to the nonuniform intensity present inside the volume of the crystal occupied by the signal beam. As stated above, the sources of the nonuniformity in our case are the crystal absorption and the Gaussian beam profiles.

As shown in Section 2, in the case of plane wave beams the shape of the gain spectrum can be predicted using the known absorption coefficient of the material (see Eq. (5)). However, an attempt to fit the frequency response data of Fig. 8(a) to Eq. (5) yields a fitted value of the absorption coefficient that differs by a factor of more than six from the actual value of 1.2 cm^{-1}. This is due to the fact that the absorption model does not account for the Gaussian nature of the pump beam or the experimental geometry. The fit to Eq. (5) is therefore somewhat artificial, in the sense that the fitting parameters are not expected to match the actual values they are meant to represent. Since the effect of non-plane-wave beams is always to reduce the frequency bandwidth, an attempt to fit the frequency response data to a model which neglects beam profiles always yields an absorption fitting parameter that equals or exceeds the actual absorption coefficient. This reflects the combined effects of the small Gaussian pump beam, the two-beam coupling geometry (including beam crossing angle and the crossing position of the beams in the crystal), and the crystal thickness.

The solid curve in Fig. 8(b) is a fit of the time response data to a single exponential rise:

$$\frac{I_{sig}(z = d/\cos\theta;\ \text{pump on})}{I_{sig}(z = d/\cos\theta;\ \text{pump off})} = e^{-\Gamma(t)d/\cos\theta} = \exp\left[C\left(1 - e^{-t/\tau}\right)\right], \qquad (7)$$

where $I_{sig}(z = d/\cos\theta;\ \text{pump on})$ is the signal beam intensity with the pump beam present, $I_{sig}(z = d/\cos\theta;\ \text{pump off})$ is the signal beam intensity with no pump beam, the signal beam is assumed to propagate in the z-direction, C is a constant that determines the steady-state gain, d is the crystal thickness, and θ is the beam crossing half-angle. Not surprisingly, these data show a correspondingly large deviation from the simple time response model and the fit to Eq. (7) yields fitting parameters that do not reflect the actual time response of the system, as it is not single exponential.

The effect on the temporal response of increasing the diameter of the Gaussian pump beam is shown in Fig. 9, in which the pump beam $1/e^2$ diameter has been increased to 10.7mm, respectively. The signal beam diameter remains at 1.8mm. Notice that the quality of the fit in Fig. 9(a) to the pure absorption model (solid curve; Eq. (5)) is extremely good, even though the fitted absorption parameter still deviates from the actual value of the absorption coefficient by a factor of 2.0. As expected, the time domain data in Fig. 9(b) show a deviation from single exponential response.

In our experiment, we had insufficient laser power to simultaneously achieve a plane wave pump beam and a large pump-signal intensity ratio. These conditions would have allowed us to extract an accurate material time constant from a fit of the frequency response data to the absorption model. Because of this experimental limitation, we took an alternative approach which consisted of illuminating the crystal incoherently from the exit face side with a large diameter beam at the same wavelength as the pump and signal beams, obtained from a separate laser. If the intensity and diameter of this "flood beam" is chosen appropriately, the effect is to significantly reduce the intensity variations in the crystal due to all sources. To the degree that intensity uniformity is accomplished, the frequency response should take on the Lorentzian shape of the zero-absorption, plane wave theory of Eqs. (1) and (2).

Figures 10 and 11 show frequency and time response data with a flood beam illuminating the crystal from the exit face side. Except for the addition of flood light, the conditions for the data of Figs. 10 and 11 were identical to those in Figs. 8 and 9, respectively. If we compare Figs. 10(a) and 8(a), for both of which the pump beam is at its smallest diameter (2.0mm), we see that the addition of the flood light in Fig. 10(a) has caused the frequency response to approach Lorentzian behavior. The remaining discrepancy is due to the fact that the pump beam intensity is more than ten times that of the flood in this case and so the flood beam is unable to completely compensate for the intensity nonuniformity in the crystal.

The greatest intensity uniformity was achieved with the largest pump beam diameter, 10.7mm, and with the flood light illuminating the crystal, the flood and pump beam intensities being approximately equal. These data are shown in Fig. 11 and from the Lorentzian and single exponential fits in Figs. 11(a) and (b) come our best estimate of the material time constant for this GaAs sample. The time constants obtained from the frequency and time domain data in Fig. 11 are in close, but not perfect agreement. This is due to the small amount of remaining intensity nonuniformity in the crystal, which is evident from the small deviation of the frequency response data from Lorentzian behavior. Note that it is impossible to completely compensate for material absorption and non-uniform beam profiles using a flood beam. However, the goodness of the fits in Figs. 11(a) and (b) and the close agreement of the time constants obtained from the frequency and time domain data (3.67(0.04) msec and 4.42(0.01) msec, respectively, at an approximate total intensity of 52 mW/cm^2) indicate that a high degree of intensity uniformity has been achieved.

It was mentioned above that the absorption model fit (Eq. (5)) of the frequency response data in Fig. 9(a), corresponding to the 10.7mm pump beam and no flood beam, yields an absorption coefficient of 2.4 cm^{-1}, which is twice the actual value. The fitted time constant also differs from the expected value based on the results in Fig. 11 and assuming a linear intensity dependence. Nevertheless, the quality of the fit to the absorption model in Fig 9(a) is very good.

The ability of the absorption model to fit the frequency response data well in almost all cases while yielding unphysical values of the absorption coefficient and time constant, is due in part to fact that the position dependent illumination, whether it originates from absorption or Gaussian beam profile, decreases the bandwidth. Thus a large absorption coefficient compensates for the beam profile effects. The other factor is the compensating relationship between the two fitting parameters. Increasing (decreasing) the value of the absorption parameter leads to a narrowing (broadening) of the gain bandwidth, while a change in the time constant parameter has the opposite effect. There is, then, a wide range of parameter pairs that yield a reasonable fit to the data. Since the absorption coefficient of the GaAs sample is known, we have fit the data of Fig. 9(a) to the absorption model with the absorption parameter fixed at its known value of 1.2 cm^{-1}. This fit is shown in Fig. 12 and the resulting fitted time constant is 7.08(0.07) msec. For comparison, the predicted value of this time constant, based on linear intensity dependence, is 6.8 msec. This is evidence that for this pump beam diameter, 10.7 mm, the influence of the Gaussian nature of the pump beam on the photorefractive temporal response has been largely eliminated.

We have observed similar two-beam coupling gain spectrum narrowing effects in four other photorefractive semiconductors: ZnTe:Mn:V, CdMnTe:V, CdTe:Ge, and CdTe:V.

4. Conclusions

In this paper, we have illustrated the influence of bulk absorption, beam profiles, and experimental geometry on measurements of photorefractive temporal response, in the undepleted pump case and in the limit of small coupling. These effects act to narrow the bandwidth and to cause the spectra to deviate from the expected Lorentzian shape.

We have shown that if accurate values of the photorefractive material time constant are to be obtained, spatial variations of total laser intensity in the crystal must either be eliminated or accounted for in the model used to extract the temporal parameters. If enough laser power is available, the pump beam can be expanded to closely approximate a plane wave. In this case, the absorption model (Eq. (5)) should describe the frequency response and a fit to this model would yield the material time constant. As we have shown, an alternative method is to achieve an approximately uniform volume intensity distribution by incoherently illuminating the material with a uniform intensity beam. Another method, in principle, is to use a very thin sample. However, in addition to depending on the availability of a thin sample having similar characteristics to the one being used in an actual system, the overall gain in this case may not be sufficient to make accurate measurements.

Acknowledgments

Financial support of EOARD and AFOSR is gratefully acknowledged.

References

1. N.V. Kukhtarev, V.B. Markov, S.G. Odulov, and V.L. Vinetskii, Ferroelectrics **22** , 949 (1979).
2. P. Yeh, *Introduction to Photorefractive Nonlinear Optics* (Wiley, New York, 1993).
3. A. Hermanns, C. Benkert, D.M. Lininger, and D.A. Anderson, IEEE J. Quant. Electron. **28**, 750 (1992).
4. D. Fluck, S. Brulisauer, and P. Gunter, Optics Commun., **115**, 626 (1995).
5. L. Solymar, D.J. Webb, and A. Grunnet-Jepsen, Prog. Quant. Electr., **18**, 377 (1994).
6. F. Vachss, in Tech. Digest, Photorefractive Materials, Effects, and Devices 1990, p140.
7. G. Brost, Opt. Commun., **96**, 113 (1993). (Note that the analysis presented here are for an applied field, but for $m=1$ the results for no applied field are similar.)

Figure Captions

FIG 1. Gain spectra for different values of αL.

FIG 2. Schematic of beam coupling interaction.

FIG 3. Calculated gain spectra for different crossing angles. Parameters were q = 1, $\alpha = 1.2\ cm^{-1}$, d = 0.6 cm, $2\omega_S$ = 2mm, $2\omega_P$ = 2mm, ξ = 1/2.

FIG 4. Calculated gain spectra for different pump beam sizes. Parameters were q = 1, $\alpha = 1.2\ cm^{-1}$, d = 0.6 cm, $2\omega_S$ = 2mm, $2\theta = 25°$, ξ = 1/2.

FIG 5. Calculated gain spectra for different beam crossing locations. Parameters were q = 1, α = $1.2\ cm^{-1}$, d = 0.6 cm, $2\omega_S$ = 2mm, $2\omega_P$ = 6mm, $2\theta = 25°$.

FIG 6. Calculated gain spectra for different values of modulation *m*.

FIG 7. Configuration for the measurements of frequency and time response of two-beam coupling. M1, M2, M3, M4: mirrors; $\lambda/2$: half-wave plate; pol: linear polarizer; BS: beamsplitter; EOM: electro-optic phase modulator; BE: variable beam expander; S: mechanical shutter; PD: photodetector; ND: neutral density filters.

FIG 8. Temporal response of the GaAs:Cr sample. $1/e^2$ beam diameters: 2.0mm (pump) and 1.8mm (signal). Spatially averaged beam intensities are $I_{pump} = 1.55\ W/cm^2$ and $I_{signal} = 2.4\ mW/cm^2$. (a) Frequency response. Solid curve: fit to the absorption model; dashed curve: fit to a Lorentzian, (b) Time response. The curve is a fit to a single exponential growth model.

FIG 9. Temporal response of the GaAs:Cr sample. $1/e^2$ beam diameters: 10.7mm (pump) and 1.8mm (signal). Spatially averaged beam intensities are $I_{pump} = 54\ mW/cm^2$ and $I_{signal} = 2.4\ mW/cm^2$. (a) Frequency response. Solid curve: fit to the absorption model; dashed curve: fit to a Lorentzian, (b) Time response. The curve is a fit to a single exponential growth model.

FIG 10. Temporal response of the GaAs:Cr sample in the presence of a flood beam. $1/e^2$ beam diameters: 2.0mm (pump) 1.8mm (signal), and 13.0mm (flood). Spatially averaged beam

intensities are $I_{pump} = 1.55\ W/cm^2$, $I_{signal} = 2.4\ mW/cm^2$, and $I_{flood} = 142\ mW/cm^2$. (a) Frequency response. Solid curve: fit to the absorption model; dashed curve: fit to a Lorentzian, (b) Time response. The curve is a fit to a single exponential growth model.

FIG 11. Temporal response of the GaAs:Cr sample in the presence of a flood beam. $1/e^2$ beam diameters: 6.0mm (pump) and 1.8mm (signal). Spatially averaged beam intensities are $I_{pump} = 54\ mW/cm^2$, $I_{signal} = 2.4\ mW/cm^2$ and $I_{flood} = 48\ mW/cm^2$. (a) Frequency response. Solid curve: fit to the absorption model; dashed curve: fit to a Lorentzian, (b) Time response. The curve is a fit to a single exponential growth model.

FIG 12. Fit of the data of Fig. 9(a) to the absorption model with the absorption coefficient parameter held constant at its known value of 1.2 cm^{-1}.

Figure 1

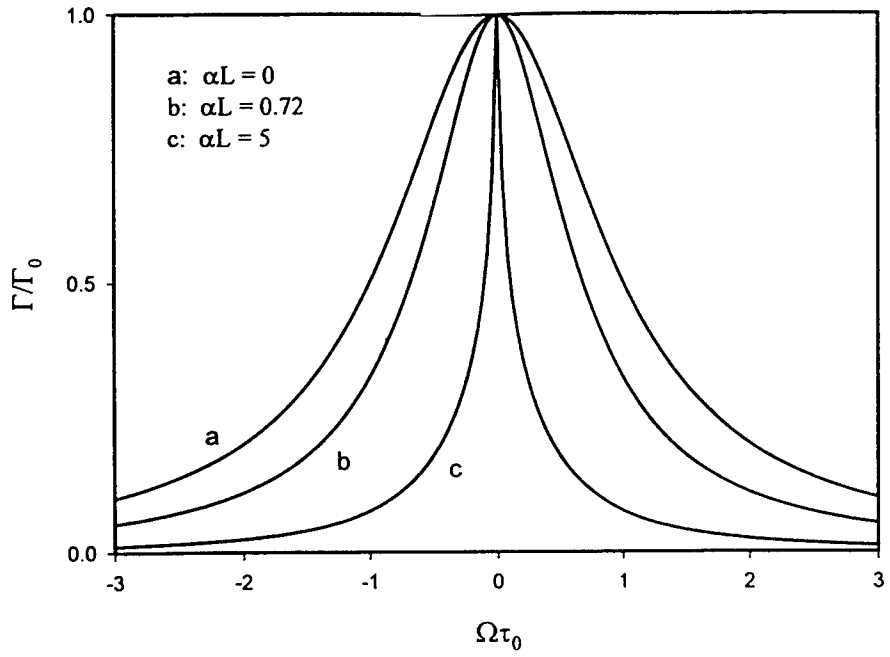


Figure 2

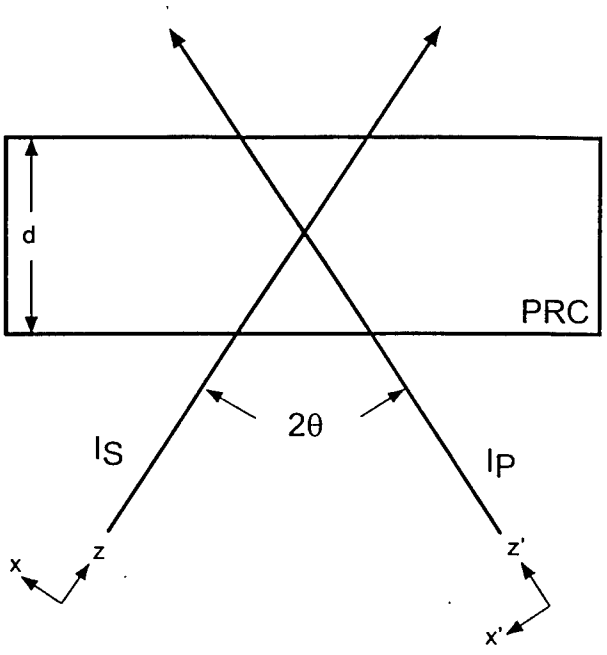


Figure 3

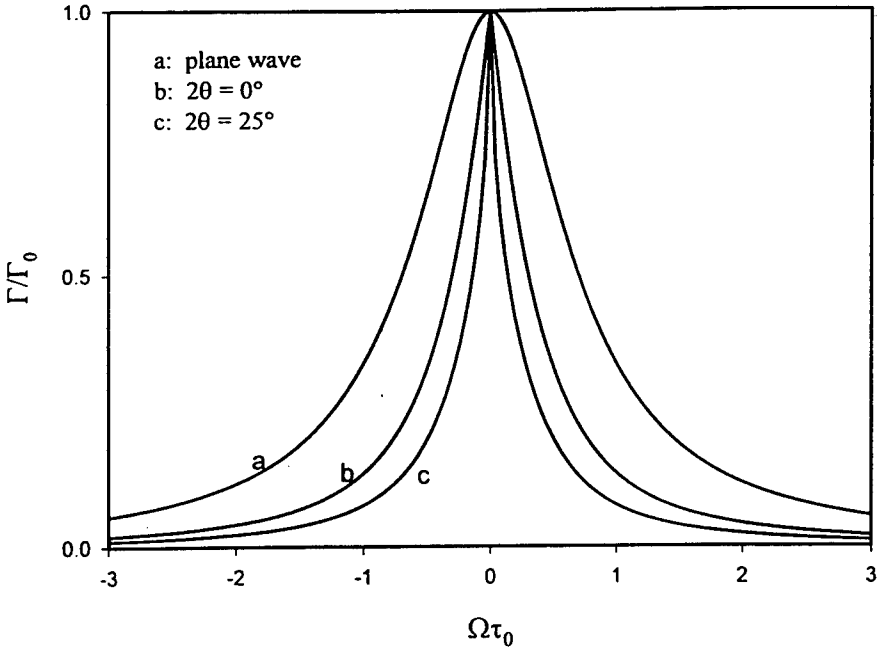


Figure 4

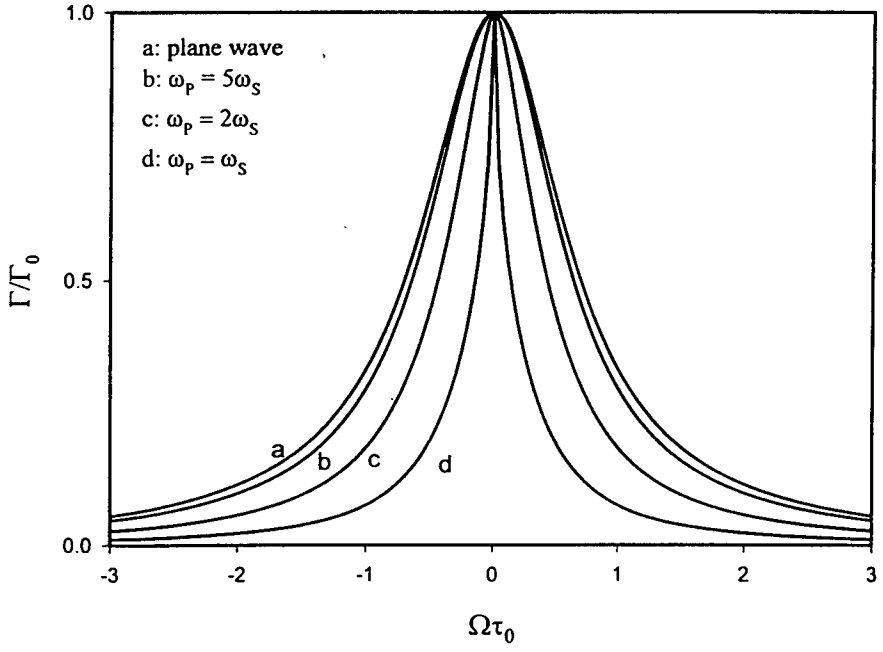


Figure 5

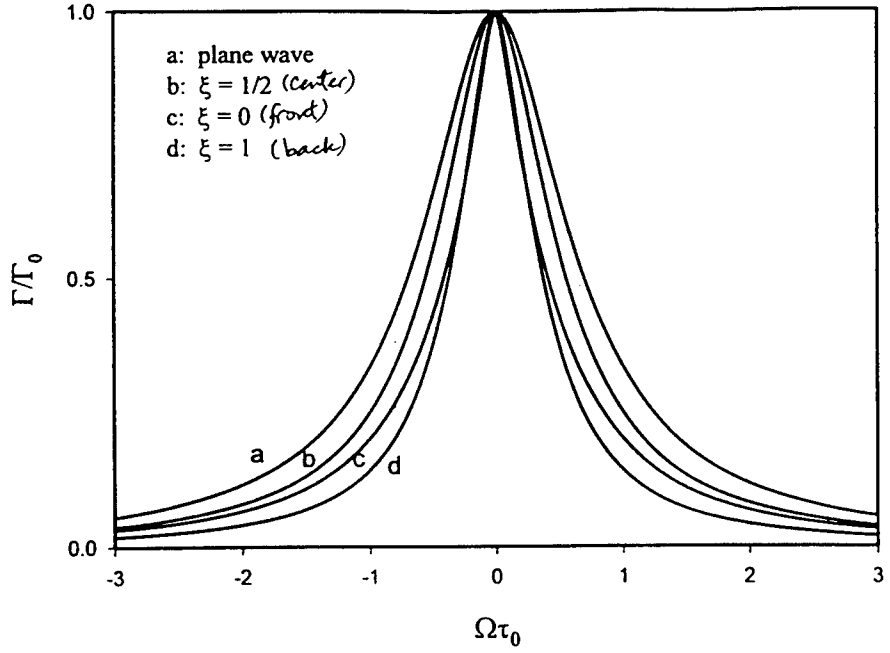


Figure 6

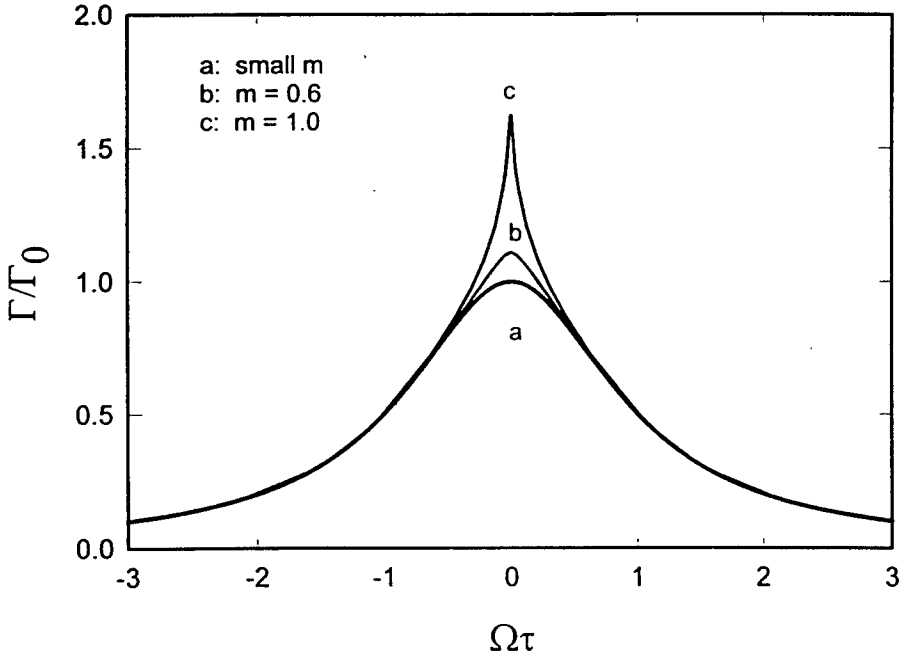


Figure 7

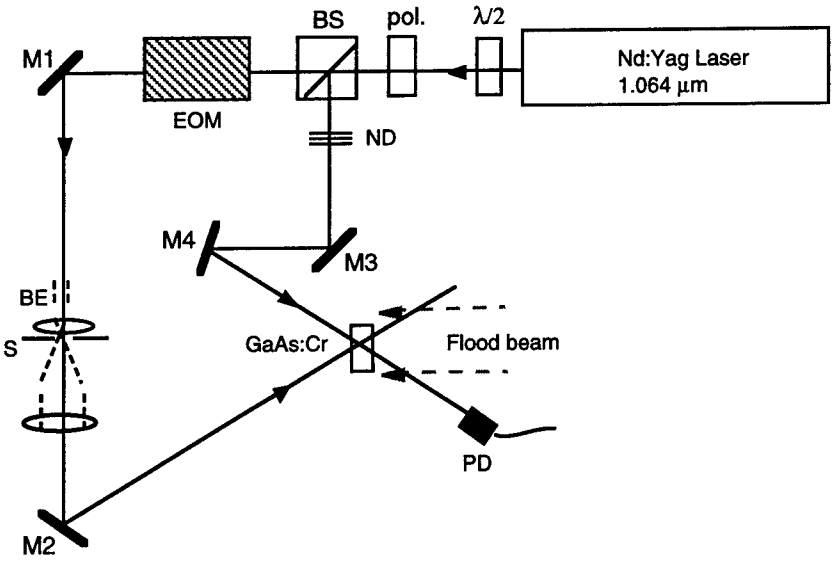

BS
pol.
λ/2
M1
Nd:Yag Laser
1.064 μm
EOM
ND
M4
M3
BE
S
GaAs:Cr
Flood beam
PD
M2

Figure 8

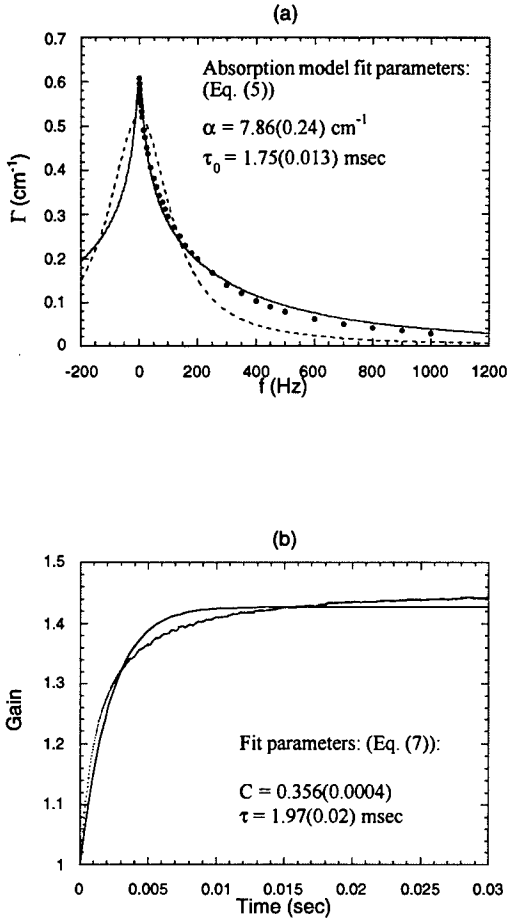


Figure 9

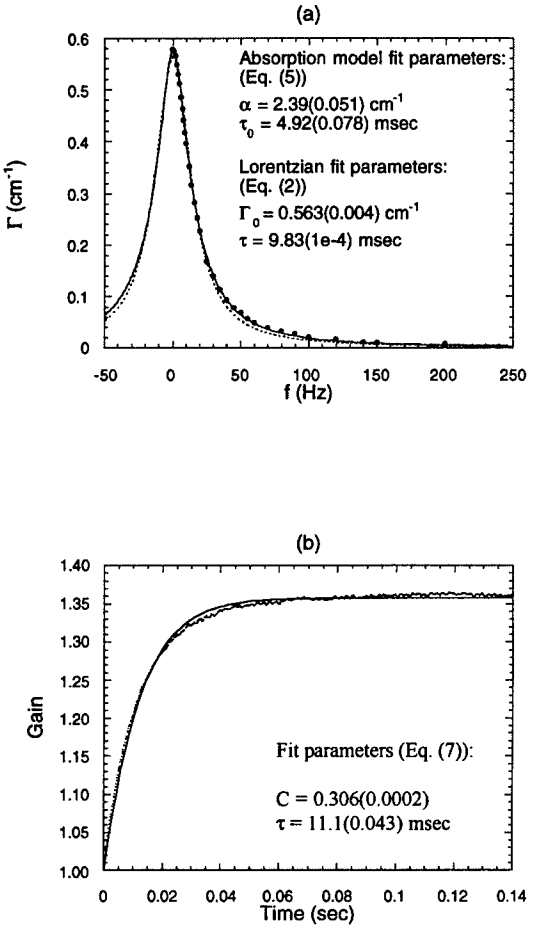


Figure 10

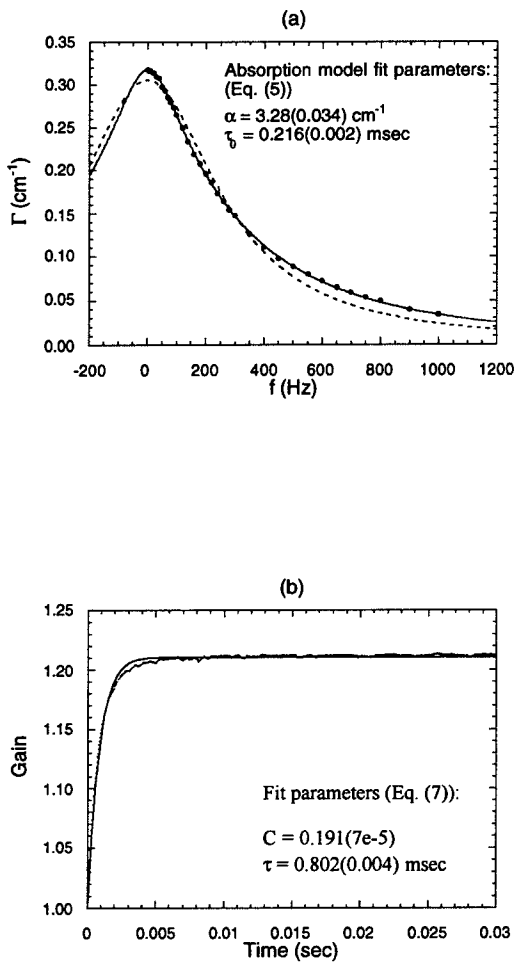


Figure 11

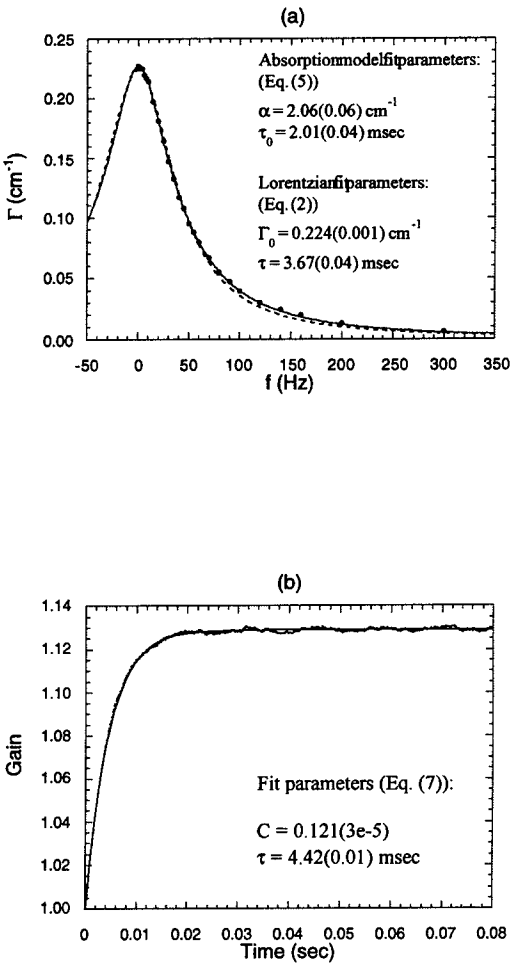
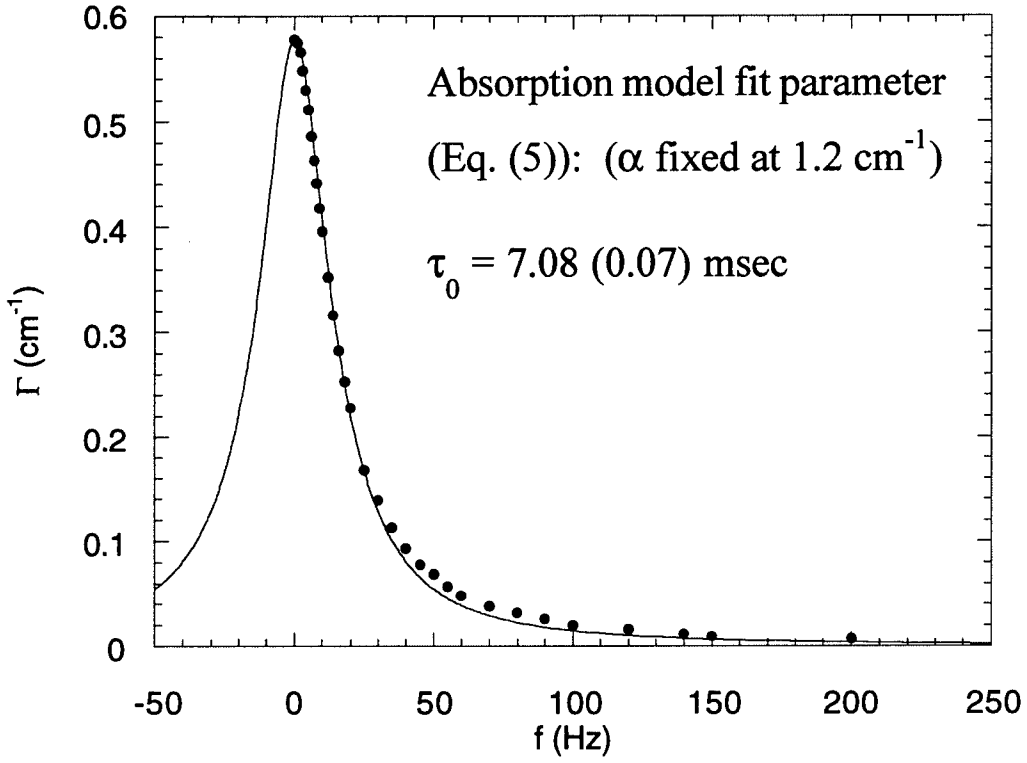


Figure 12


Absorption model fit parameter
(Eq. (5)): (α fixed at 1.2 cm-1)
τ0 = 7.08 (0.07) msec
Γ (cm-1)
f (Hz)
0.6
0.5
0.4
0.3
0.2
0.1
0
-50
0
50
100
150
200
250

Associate did not participate in program.

JOINT DOMAIN SPACE-TIME ADAPTIVE PROCESSING WITH SMALL TRAINING DATA SETS

Dimitris A. Pados
Associate Professor
Department of Electrical and Computer Science

State University of New York at Buffalo
201 Bell Hall
Buffalo, NY 14260

Final Report for:
Summer Research Program
Rome Laboratory

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, Washington, DC

And

Rome Laboratory

August 1997

JOINT DOMAIN SPACE-TIME ADAPTIVE PROCESSING WITH SMALL TRAINING DATA SETS

Dimitris A. Pados
Assistant Professor
Department of Electrical and Computer Engineering
State University of New York at Buffalo
E-mail: pados@eng.buffalo.edu

Abstract

The classical problem of optimum detection of a signal of unknown amplitude in Gaussian noise is revisited. The focus, however, is on adaptive system designs through limited training sets. Traditionally, adaptive detectors involve the inverse of the sampled covariance matrix of the noise process. In this work, linear filter optimization is carried out on a complex hyperplane and in the adaptive form no sample matrix inversion is necessary. This results in computational savings of an order of magnitude and superior probability of detection performance for small training set instances. An interesting by-product of the algorithm developed herein is significant test resistance when training data include the signal of interest.

While the issues treated refer to general adaptive detection procedures, the presentation is given in the context of joint space-time adaptive processing for array radars.

JOINT DOMAIN SPACE-TIME ADAPTIVE PROCESSING WITH SMALL TRAINING DATA SETS

Dimitris A. Pados

I. Introduction

A hypothesis testing problem is called general Gaussian if the data is Gaussian distributed under all hypotheses [1]. Detection of a signal in Gaussian noise falls under the binary hypothesis testing version of the general Gaussian problem and lies in the foundation of the theory of optimum detection. Optimum data processing in the Bayes, Neyman-Pearson, maximum output Signal-to-Noise-Ratio (SNR), minimum Mean-Square-Error (MSE), or Minimum-Variance-Distortionless-Response (MVDR) sense are all well known, and lead to scaled versions of the same linear filter solution [1], [2], traditionally called Wiener or colored noise Matched Filter (MF). Calculation of this filter requires knowledge of the inverse covariance matrix of the noise process.

In the context of an array radar application with M antenna elements (spatial channels) and N pulses per coherent processing interval (CPI), optimum signal detection in the presence of spatial and temporal Gaussian noise requires joint space-time matched filtering in the $M \times N$ complex vector space [3]. Since prior knowledge of the noise covariance matrix is not available, the maximum-likelihood sample average estimate is often used, developed from K noise only vector samples that correspond to distinct range cells. This is the so called secondary data set. Then, the inverse of the sample-matrix is considered as an estimate of the inverse covariance matrix. This approach is known as the Sample-Matrix-Inversion method (SMI) and in [4] it was shown to outperform the recursive least-mean-squares (LMS) adaptive implementation of the matched filter processor in terms of convergence rate and small-sample output SNR characteristics. Still, it was found that $K \geq 2MN$ independent and identically distributed (i.i.d.) training data samples are needed to maintain with probability 1/2 a loss lower than or equal to 3dB compared to the ideal MF. Simple variance normalization of the MF

decision statistic led to a Constant False Alarm Rate (CFAR) test [5] for Neyman-Pearson detection.

System optimization in the generalized likelihood ratio (GLR) sense was pursued in [6]. The resulting test statistic offers embedded CFAR behavior, converges to the ideal MF solution at least in probability as the number of secondary data K grows, and it is shown in [5] to outperform the SMI approach for $K = 2MN$, except for high SNR regions. Extended performance analysis and studies of the sidelobe behavior of the GLR filter are carried out in [7].

It is important to note that the SMI and the GLR test are both asymptotically optimum to the extent that they converge in a probabilistic sense to the optimum ideal MF as the size of the secondary data set grows to infinity. However, for finite sample support no optimality can be claimed in either case and superior probability of detection performance for a fixed false alarm rate is theoretically possible by other filtering means. Moreover, both methods share the need to invert the sample covariance matrix of the noise process which implies: *(i)* computational complexity of order $(MN)^3$, *(ii)* secondary data set of at least $K = 2MN$ samples to maintain with probability $1/2$ a loss compared to the optimum rule of no more than 3dB, and *(iii)* $K > MN$ to ensure nonsingularity of the sample covariance matrix with probability 1 [6]. The latter is due to characteristics of the Wishart distribution that the sample covariance matrix is assumed to follow under the i.i.d. and Gaussian assumptions.

Arguably, in airborne surveillance systems the training data size requirements make the practicality of these approaches questionable even for moderate values of M and N. This is particularly true if we consider a highly non-stationery, non-homogeneous [8] operating environment, typically encountered in practice. In this context, it is tempting to consider abolishing the concept of joint space-time processing and pursue separate ("disjoint" or "cascade") SMI or GLR detection in the M-dimensional space and the N-dimensional time domain. It is easy, however, to identify realistic cases where the performance of these cascade schemes falls unacceptably below that of the joint domain approaches [9].

In this present work we preserve the principle of joint space-time processing but instead of seeking an adaptive implementation of the optimum linear filter in the C^{MN} complex space, we pursue a suboptimum (simpler) solution. We optimize adaptively a linear filter on a carefully selected complex

hyperplane in the maximum output Signal-to-Interference-plus-Noise-Ratio (SINR) sense. As a result, the decision statistic requires no inversion of the sample covariance matrix. Computational savings of an order of magnitude are gained, but most importantly superior performance in terms of probability of detection is exhibited for small size $K \leq 2MN$ secondary data sets as compared to the methods that involve sample matrix inversion in the C^{MN} space (SMI and GLR). Meaningful system optimization can be carried out even if $K < MN$, where the sample covariance matrix is singular. By construction, another valuable feature of this new decision statistic is significant resistance and performance stability when system adaptation is carried out with a secondary data set corrupted by the signal of interest. This is known in the radar literature as a "multiple interfering target" situation or "raid formation". Numerical and simulation results included in this work support the theoretical arguments and promote the new method as a desirable alternative for joint space-time adaptive processing when very small secondary data training sets are available.

The rest of this report is organized as follows. Section II introduces the system and signal models. The principles of Auxiliary-Vector filtering and the new test statistic are developed and studied in Section III. Probability of detection comparisons with the ideal Matched-Filter, the adaptive Matched-Filter and the Generalized-Likelihood-Ratio test are carried out in Section IV for a few selected cases. Concluding remarks are drawn in Section V.

II. Signal Model and Background

Consider a narrowband uniform linear radar array with M antenna elements (subarrays or spatial channels). Also, assume that each element collects the complex (I/Q) return of a series of N coherent pulses for some given range cell $k = 1, \ldots, K_{max} = T_{PRI}/T$, where T_{PRI} is the pulse repetition interval and T is the pulse duration. We organize the received data in the form of a matrix $X_{M\times N}$, where $X(m,n)$, $m = 1, \ldots, M$, $n = 1, \ldots, N$, denotes the m-element, n-pulse signal sample. The objective is to cope with system and surrounding disturbances and detect in $X_{M\times N}$ the presence of a desired signal of unknown amplitude. Without loss of generality and for notational simplicity, we consider a "vectorized" form of $X_{M\times N}$, where $\text{Vec}(X_{M\times N}) = X_{MN\times 1}$ is constructed by sequencing all matrix

columns in the form of a vector. From now on, bold variables indicate vectors in the C^{MN} complex space unless otherwise specified.

We begin by casting the detection problem in the context of binary hypothesis testing. We denote the disturbance only hypothesis by H_0 and the target plus disturbance hypothesis by H_1.

$$\begin{aligned} \mathrm{H_0}: \mathbf{X} &= \mathbf{J}+\mathbf{C}+\mathbf{N} \\ \mathrm{H_1}: \mathbf{X} &= \alpha\mathbf{V}+\mathbf{J}+\mathbf{C}+\mathbf{N}. \end{aligned} \tag{1}$$

In (1), $\mathbf{J}$ represents a mixture of L broadband directional interferers (jammers) in the far field, where $\mathbf{J} = Vec(J_{M\times N})$ and

$$J(m,n) = \sum_{l=1}^{L} J_l(n)\, e^{j2\pi(m-1)\frac{sin\theta_l d}{\lambda}},\ n = 1,\ldots,N,\ m = 1,\ldots,M. \tag{2}$$

We assume that $J_l(n), l = 1,\ldots,L$, is complex white Gaussian distributed to account for channel fading at the pulse-repetition frequency as in the model of [10] and [11]. The antenna element spacing is d and the radar carrier wavelength is λ. The direction of arrival (DOA) $\theta_l, l = 1,\ldots,L$, is assumed to be uniformly distributed in $[-\pi/2, \pi/2]$. We find it convenient to define the spatial frequency

$$f_l \triangleq \frac{sin\theta_l d}{\lambda},\ l = 1,\ldots,L, \tag{3}$$

and assume that f_l is uniformly distributed in [-0.5,0.5] with proper selection of d and λ. In addition, $\mathbf{C}_{MN\times 1}$ in (1) accounts for colored Gaussian noise with covariance matrix $\mathbf{R}_c$ and corresponds to a radar clutter region. Spatially and temporally white disturbances are denoted by $\mathbf{N}$. The signal or "steering vector" of interest, $\mathbf{V}$, is present in $\mathbf{X}$ under hypothesis H_1 only. Without loss of generality we assume that $\mathbf{V}^H\mathbf{V} = 1$ such that all energy signal characteristics are absorbed in the unknown complex amplitude constant α ($\mathbf{V}^H$ denotes the Hermitian operation on $\mathbf{V}$). For completeness, if $\mathbf{V} = Vec(V_{M\times N})$ then

$$V(m,n) = \frac{1}{\sqrt{MN}}\, e^{j2\pi(m-1)\frac{sin\theta_s d}{\lambda} + j2\pi(n-1)\frac{2\nu}{\lambda f_{PR}}}. \tag{4}$$

In (4), $\theta_s \in [-\pi/2, \pi/2]$ is the angle of arrival of the signal (target) of interest, ν is the target radial velocity, and f_{PRF} is the radar pulse repetition frequency. Once again, it is convenient to define the

spatial target frequency

$$f_s \triangleq \frac{sin\theta_s d}{\lambda} \tag{5}$$

and the"normalized Doppler" target frequency

$$f_D \triangleq \frac{2\nu}{\lambda f_{PR}}. \tag{6}$$

In this case we assume that $f_s, f_D \in [-0.5, 0.5]$.

Given a data vector $\mathbf{X}$ that corresponds to some range cell $k \in \{1, \ldots, K_{max}\}$, the objective is to decide in favor of H_0 or H_1 in a way that maximizes the probability of detection $P_D = Pr\{H_1 \; decided | H_1 \; true\}$ subject to a given false alarm constraint $P_{FA} = Pr\{H_1 \; decided | H_0 \; true\} \leq \rho$. The optimum decision rule is well known [1], [3] and of the form

$$|\mathbf{w}^H \mathbf{X}| \underset{H_0}{\overset{H_1}{\gtrless}} \tau, \tag{7}$$

where $\tau > 0$ is the threshold parameter to be determined according to the condition $P_{FA} = \rho$ and $\mathbf{w}$ is the linear filter defined by

$$\mathbf{w} = \mathbf{R}^{-1}\mathbf{V}. \tag{8}$$

In (8), $\mathbf{R} = E_{H_0}\{\mathbf{X}\mathbf{X}^H\}$ where $E_{H_i}\{\cdot\}$ denotes the statistical expectation operation under $H_i, i = 0, 1$. Then the variance under H_0 of the test statistic $\mathbf{w}^H\mathbf{X}$ is $Var_{H_0}\{\mathbf{w}^H\mathbf{X}\} = \mathbf{V}^H\mathbf{R}^{-1}\mathbf{V}$ which implies that the modified test statistic $\frac{\mathbf{V}^H\mathbf{R}^{-1}\mathbf{X}}{\sqrt{\mathbf{V}^H\mathbf{R}^{-1}\mathbf{V}}}$ is zero-mean unit-variance complex Gaussian distributed. As a consequence $\left|\frac{\mathbf{V}^H\mathbf{R}^{-1}\mathbf{X}}{\sqrt{\mathbf{V}^H\mathbf{R}^{-1}\mathbf{V}}}\right|^2$ is chi-square distributed with two degrees of freedom and leads to the familiar CFAR (constant false alarm rate) optimum decision rule

$$\frac{|\mathbf{V}^H\mathbf{R}^{-1}\mathbf{X}|^2}{\mathbf{V}^H\mathbf{R}^{-1}\mathbf{V}} \underset{H_0}{\overset{H_1}{\gtrless}} \lambda, \qquad (\lambda > 0). \tag{9}$$

Substitution in (9) of the sample-average covariance matrix estimate $\hat{\mathbf{R}} = \frac{1}{K}\sum_{k=1}^{K} \mathbf{X}_k\mathbf{X}_k^H$ from K data samples from H_0 defines the so called CFAR adaptive matched filter (AMF) detector [5]. In

contrast, we recall that the generalized likelihood ratio (GLR) test of [6] and [7] is

$$\frac{|\mathbf{V}^H\hat{\mathbf{R}}^{-1}\mathbf{X}|^2}{\mathbf{V}^H\hat{\mathbf{R}}^{-1}\mathbf{V}(1+\frac{1}{K}\mathbf{X}^H\hat{\mathbf{R}}^{-1}\mathbf{X})} \underset{H_0}{\overset{H_1}{\gtrless}} \gamma, \qquad (\gamma > 0). \tag{10}$$

In the following section we develop a new decision statistic that maintains the principle of linear filtering for signal detection in Gaussian disturbance. However, filter optimization is carried out on a complex hyperplane instead of the whole C^{MN} space. As a result, no sample matrix inversion is necessary, computational gains of an order of magnitude are achieved compared to the AMF and GLR approaches, and superior performance is exhibited for optimization with small secondary data sets ($K < 2MN$). In fact, test optimization can be pursued even if $K < MN$. By construction, the test is resistant to training with samples that contain the target of interest.

III. Adaptive Auxiliary-Vector Detection

Let us consider the problem of identifying the filter $\mathbf{w} \in C^{MN}$ that minimizes the output variance under hypothesis H_0, $E_{H_0}\{|\mathbf{w}^H\mathbf{X}|^2\}$, subject to the constraint that $\mathbf{w}^H\mathbf{V} = 1$, where $\mathbf{V}$ is the target signal of interest. The solution to this problem is well known to be [12]

$$\mathbf{w}_{MVDR} = \frac{\mathbf{R}^{-1}\mathbf{V}}{\mathbf{V}^H\mathbf{R}\mathbf{V}} \tag{11}$$

and it is frequently cited as the Minimum-Variance-Distortionless-Response (MVDR) filter. We observe that $\mathbf{w}_{MVDR}$ defines a scaled, hence equivalent, version of the optimum decision rule in (7) and (8). Therefore, without loss of generality, we may seek the optimum decision statistic within the class of filters $\mathbf{w}$ such that [14]

$$\mathbf{w} = \mathbf{V} - \mu\mathbf{G}, \tag{12}$$

where $\mu \in C$ and $\mathbf{G}$ is an arbitrary vector in C^{MN} orthonormal to $\mathbf{V}$; that is,

$$\mathbf{G}^H\mathbf{V} = 0 \qquad \text{and} \qquad \mathbf{G}^H\mathbf{G} = 1. \tag{13}$$

We identify $\mathbf{G}$ as the "auxiliary" vector and refer to μ as the "steering" (complex scalar) filter parameter. The objective is to select $\mathbf{G}$ appropriately and then optimize the filter with respect to μ.

We begin with the observation that all interference/clutter vectors in the vector direction of $\mathbf{G} + \mu^*\mathbf{V}$ are completely rejected, since $[\mathbf{V} - \mu\mathbf{G}]^H c[\mathbf{G} + \mu^*\mathbf{V}] = 0 \ \forall c \in C$. Here μ^* denotes the complex conjugate of μ. Moreover, the filter output for an H_1 data instance (target present) is:

$$[\mathbf{V} - \mu\mathbf{G}]^H \mathbf{X}_{|H_1} = \alpha + \mathbf{V}^H(\mathbf{J} + \mathbf{C}) - \mu^*\mathbf{G}^H(\mathbf{J} + \mathbf{C}) + [\mathbf{V} - \mu\mathbf{G}]^H \mathbf{N}. \tag{14}$$

Assuming that interference, clutter, and white noise are independent, the output variance under H_1 equals

$$E_{H_1}\{|[\mathbf{V} - \mu\mathbf{G}]^H \mathbf{X}|^2\} = |\alpha|^2 + [\mathbf{V} - \mu\mathbf{G}]^H(\mathbf{R}_J + \mathbf{R}_C)[\mathbf{V} - \mu\mathbf{G}] + (1 + |\mu|^2)\sigma^2, \tag{15}$$

where $\mathbf{R}_J = E\{\mathbf{J}\mathbf{J}^H\}, \mathbf{R}_C = E\{\mathbf{C}\mathbf{C}^H\}$, and σ^2 is assumed to be the variance of the spatially and temporally white Gaussian noise. From (14) we see that if $\mu^* = \frac{\mathbf{V}^H(\mathbf{J}+\mathbf{C})}{\mathbf{G}^H(\mathbf{J}+\mathbf{C})}$ then the interference/clutter contribution is eliminated for that input instance, while (15) shows that the output white noise variance contribution increases linearly in $|\mu|^2$. We seek an auxiliary vector $\mathbf{G}$ that satisfies the conditions in (13) and is close to the average $\mathbf{J} + \mathbf{C}$ line direction in the Euclidean sense. If we define

$$\mathbf{X}_{\perp \mathbf{r}} \stackrel{\triangle}{=} sgn(Re[\mathbf{V}^H\mathbf{X}]) \frac{\mathbf{X} - \mathbf{V}^H\mathbf{X}\mathbf{V}}{\|\mathbf{X} - \mathbf{V}^H\mathbf{X}\mathbf{V}\|}, \quad \text{then} \tag{16}$$

$$\mathbf{G} \stackrel{\triangle}{=} \frac{E_{H_0}\{\mathbf{X}_{\perp \mathbf{r}}\}}{\|E_{H_0}\{\mathbf{X}_{\perp \mathbf{r}}\}\|}. \tag{17}$$

The following proposition identifies the complex scalar μ that maximizes the output SINR (Signal-to-Interference-plus-Noise) for any given vector $\mathbf{G}$. The proof is omitted due to lack of space.

Proposition 1 *(i) For a filter structure as in (12) and any auxiliary vector* $\mathbf{G}$ *that satisfies (13), the complex scalar* μ *that maximizes the output SINR (minimizes* $E_{H_0}\{|[\mathbf{V} - \mu\mathbf{G}]^H\mathbf{X}|^2\}$*) is*

$$\mu = \frac{\mathbf{G}^H\mathbf{R}\mathbf{V}}{\mathbf{G}^H\mathbf{R}\mathbf{G}}. \tag{18}$$

(ii) For the optimum μ *in (18) the filter output variance becomes*

$$E_{H_0}\{|[\mathbf{V} - \mu\mathbf{G}]^H\mathbf{X}|^2\} = \mathbf{V}^H\mathbf{R}\mathbf{V} - \frac{|\mathbf{G}^H\mathbf{R}\mathbf{V}|^2}{\mathbf{G}^H\mathbf{R}\mathbf{G}}. \square \tag{19}$$

In view of this result, the filter under consideration in this work is $\mathbf{w}_{AV} = \mathbf{V} - \mu\mathbf{G}$ with $\mathbf{G}$ identified by (16) and (17) and μ given by (18). The approximation error compared to the optimum

distortionless filter in (11) can be quantified in terms of the second order moment (variance) under H_0. We can calculate

$$E_{H_0}\{|\,\mathbf{w}_{AV}^H\mathbf{X}\,|^2\} - E_{H_0}\{|\,\mathbf{w}_{MVDR}^H\mathbf{X}\,|^2\} = \mathbf{V}^H\mathbf{R}\mathbf{V} - \frac{|\,\mathbf{G}^H\mathbf{R}\mathbf{V}\,|^2}{\mathbf{G}^H\mathbf{R}\mathbf{G}} - \frac{1}{\mathbf{V}^H\mathbf{R}^{-1}\mathbf{V}} \geq 0. \tag{20}$$

For severely ill-conditioned matrices $\mathbf{R}$ with high eigenvalue spread [12], [13] the approximation error may be significant. Still, for adaptive filter implementations through small secondary data sets ($K \sim MN$), the computational savings that come with the application of the auxiliary-vector filter are coupled by strong performance gains compared to the adaptive MF or GLR approaches. We will return to this issue in the next section where we present two illustrative examples.

The output variance of the filter $\mathbf{w}_{AV}$ under H_0 is $\mathbf{w}_{AV}^H\mathbf{R}\mathbf{w}_{AV}$, which implies that the decision statistic $\frac{\mathbf{w}_{AV}\mathbf{X}}{\sqrt{\mathbf{w}_{AV}^H\mathbf{R}\mathbf{w}_{AV}}}$ is *0*--mean, unit-variance complex Gaussian. Then,

$$\frac{|\,\mathbf{w}_{AV}\mathbf{X}\,|^2}{\mathbf{w}_{AV}^H\mathbf{R}\mathbf{w}_{AV}} \underset{H_0}{\overset{H_1}{\gtrless}} \lambda \tag{21}$$

defines a CFAR test in parallel to the optimum matched filter in (9). In adaptive implementations of $\mathbf{w}_{AV}$, say $\hat{\mathbf{w}}_{AV}$, the statistical expectation operation for the $\mathbf{G}$ expression in (17) is substituted by sample-averaging:

$$\hat{\mathbf{G}} = \frac{\sum_{k=1}^{K}\mathbf{X}_{\perp V}(k)}{\|\sum_{k=1}^{K}\mathbf{X}_{\perp V}(k)\|}. \tag{22}$$

The covariance matrix $\mathbf{R}$ is needed in (18) for the evaluation of the μ scalar and in (21) for the test variance normalization. It is estimated through the maximum likelihood sample-average procedure as usual:

$$\hat{\mathbf{R}} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{X}(k)\mathbf{X}(k)^H. \tag{23}$$

Then, the adaptive auxiliary-vector test becomes

$$\frac{|\,\hat{\mathbf{w}}_{AV}\mathbf{X}\,|^2}{\hat{\mathbf{w}}_{AV}^H\hat{\mathbf{R}}\hat{\mathbf{w}}_{AV}} \underset{H_0}{\overset{H_1}{\gtrless}} \hat{\lambda}, \tag{24}$$

where $\hat{\mathbf{w}}_{AV}$ is identified through (16), (22), (18), and (23). The test is assymptotically CFAR (as $K \to \infty$, $\hat{\lambda} \to \lambda$ in (9) for the optimum normalized matched-filter test for any given false alarm rate

P_{FA}). For finite training sets it qualifies as Cell-Averaging CFAR (CA-CFAR), as seen by the test denominator.

IV. Numerical and Simulation Comparisons

In this section we support and illustrate the preceding theoretical developments through two case-studies based on the hypothesis testing problem in (1). In all cases we assume presence of a mixture of 3 (three) broadband interferers with Jammer-to-Noise-Ratio $JNR = \frac{E\{|\mathbf{J}_l|^2\}}{\sigma^2} \approx 35dB$, $l = 1,2,3$. The jammers follow the model in (3) and the spatial frequency f_l, $l = 1,2,3$, is randomly drawn from the uniform $[-.5,.5]$ distribution. The "peak" clutter (colored Gaussian noise) to noise ratio is fixed at $CNR = \frac{\sigma_c^2}{\sigma^2} = 20dB$. The false alarm rate is set at $P_{FA} = .01$ and for simulation purposes threshold and probability of detection estimates P_D are based on $10,000$ samples from H_0 and H_1 respectively. All presented results are averages over 64 independent Monte-Carlo runs for arbitrarily chosen target vectors $\mathbf{V}$. P_D values are plotted as a function of the total SINR defined by

$$SINR = |\alpha|^2 \mathbf{V}^H \mathbf{R}^{-1} \mathbf{V}. \tag{25}$$

All graphs include the ideal matched filter as a reference point and the adaptive matched filter, GLR, and the Auxiliary-Vector test for various secondary data set sizes. We assume a radar array system with $M = 5$ antenna elements (channels) and $N = 12$ pulses per coherent processing interval (CPI). The number of available training data is denoted by K.

First, we consider performance studies when the clutter disturbance is modeled by a colored Gaussian process with a dense and "well-behaved" covariance matrix $\mathbf{R}_{c_1}$. We choose, for example, a Toeplitz matrix where the first $MN \times 1 (60 \times 1)$ column is defined by $R_{c_1}(1,1) = 100$, $R_{c_1}(i,1) = (62-i)+j$ for $i = 2,\ldots,20$, $R_{c_1}(i,1) = (71-i)+j$ for $i = 21,\ldots,40$, $R_{c_1}(i,1) = (81-i)+j$ for $i = 41,\ldots,60$. If $e_1, e_2, \ldots, e_{60}$ denote the eigenvalues of $\mathbf{R}_{c_1}$ in a descending order, then the eigenvalue spread is $\chi(\mathbf{R}_{c_1}) = \frac{e_1}{e_{60}} \approx 80$ and $\frac{e_1}{e_2} \approx 6.18$. Fig. 1 shows the probability of detection P_D as a function of the SINR. The adaptive schemes utilize a secondary data set of $K = 2MN = 120$ samples. In Fig. 2 we present the same studies but now only $K = MN + 1 = 61$ training samples are assumed available. The auxiliary-vector filter design for $K = 30 < MN$ is also included in this figure.

Next we consider an extreme ill-conditioned covariance matrix case to magnify the effects of the approximation error in (20). We choose the sparse Toeplitz matrix $\mathbf{R}_{c_2}$ with the first column elements as follows: $R_{c_2}(1,1) = 100$, $R_{c_2}(2,1) = R_{c_2}(6,1) = 53.94j$, $R_{c_2}(3,1) = R_{c_2}(11,1) = -8.47$, $R_{c_2}(4,1) = R_{c_2}(16,1) = -0.39j$, $R_{c_2}(5,1) = R_{c_2}(13,1) = R_{c_2}(17,1) = R_{c_2}(23,1) = 0.01$, $R_{c_2}(7,1) = -9$, $R_{c_2}(8,1) = R_{c_2}(12,1) = -0.44j$, and $R_{c_2}(i,1) = 0$ for all other $i \in \{1,\ldots,60\}$. The eigenvalue spread for this matrix is $\chi(\mathbf{R}_{c_2}) = \frac{e_1}{e_{60}} >> 10^5$ and $\frac{e_1}{e_2} \approx 1.1$. Fig. 3 repeats the studies of Fig. 2 for colored Gaussian noise with covariance matrix $\mathbf{R}_{c_2}$. While the P_D for the AV test degrades significantly compared to the previous $\mathbf{R}_{c_1}$ case, the AV test still outperforms the GRL test by about 3dB while the performance of the adaptive MF test is even lower.

Finally, we repeat the studies of Fig. 1, Fig. 2, and Fig. 3, but now we also account for multiple target interference (known as "raid formation" interference in the radar literature). In this context, we assume that about 10% of the training data include the target of interest (that is, they come from hypothesis H_1). Twelve, six, and three data samples include the target when training is carried out with 120, 61, and 30 samples respectively. The probability of detection versus SINR results are given in Figs. 4, 5 and 6 that parallel the plots in Figs. 1, 2, and 3. Severe performance degradation is shown for the GLR and the adaptive MF test, while the Auxiliary-Vector approach exhibits significant resistance.

V. Conclusions

We have developed an adaptive decision rule for the detection of signals of unknown amplitude in Gaussian noise. Linear filter optimization is carried out on a complex hyperplane and the adaptive form does not require the inverse of the sampled covariance matrix of the noise process. The significant computational savings are apparent, but the actual objective is in fact improved probability of detection performance for system adaptation with small training sets.

Fig. 1 and Fig. 2 of the previous section accounted for probability of detection studies for optimistic and well-behaved colored noise covariance matrices. Still, they show how far from optimum the conventional AMF and GLR methods may be. Fig. 3 presented an extreme ill-conditioned covariance

matrix case and the superiority of the proposed approach is maintained for secondary set sizes near the joint space-time dimensionality ($K \approx MN$). Since no matrix inversion is performed system adaptation may be pursued even if $K \leq MN$, where the covariance matrix estimate is singular.

We also note that the proposed filter decomposition into the signal of interest and an orthonormal component (auxiliary vector) results to relative performance stability when the secondary data set includes target "contaminated" samples.

References

[1] H. L. Van Trees, *Detection, Estimation and Modulation Theory, Part I* , New York, NY: Wiley, 1968.

[2] V. Solo and X. Kong, *Adaptive Signal Processing Algorithms*, Englewood Cliffs, NJ: Prentice Hall, 1995.

[3] L. E. Brennan and I. S. Reed, "Theory of adaptive radar," *IEEE Trans. Aerospace and Electr. Syst.*, vol. AES-9, pp. 237-252, Mar. 1973.

[4] L. S. Reed, J. D. Mallet, and L. E. Brennan, "Rapid convergence rate in adaptive arrays," *IEEE Trans. Aerospace and Electr. Syst.*, vol. AES-10, pp. 853-863, Nov. 1974.

[5] F. C. Robey, D. R. Fuhrmann, E. J. Kelly, and R. Nitzberg, "A CFAR adaptive matched filter detector," *IEEE Trans. Aerospace and Electr. Syst.*, vol. AES-28, pp. 208-216, Jun. 1992.

[6] E. J. Kelly, "An adaptive detection algorithm," *IEEE Trans. Aerospace and Electr. Syst.* , vol. AES-22, pp. 115-127, Mar. 1986.

[7] E. J. Kelly, "Performance of an adaptive detection algorithm; Rejection of unwanted signals," *IEEE Trans. Aerospace and Electr. Syst.*, vol. AES-25, pp. 122-133, Mar. 1989.

[8] W. L. Melvin and M. C. Wicks, "Improving practical space-time adaptive radar," in *Proc. of 1997 IEEE National Radar Conf.*, pp. 48-53, Syracuse, NY, May 1997.

[9] H. Wang and L. Cai, "On adaptive spatial-temporal processing for airborne surveillance radar systems," *IEEE Trans. Aerospace and Electr. Syst.*, vol. AES-30, pp. 660-670, Jul. 1994.

[10] J. R. Roman and D. W. Davis, "Multichannel system identification and detection theory using output data techniques: Volume II", USAF Rome Laboratory, report No. SSC-TR-96-02, Oct. 1996.

[11] J. Ward, "Space-time adaptive processing for airborne radar", MIT Lincoln Laboratory, Technical Report No. TR-1015, Dec. 1994.

[12] S. Haykin, *Adaptive Filter Theory.* Englewood Cliffs, NJ: Prentice Hall, 2nd ed., 1991.

[13] R. A. Horn and C. R. Johnson, *Matrix Analysis.* New York, NY: Cambridge University Press, 5th ed., 1991.

[14] D. A. Pados and S. N. Batalama, "Low complexity blind detection of DS/CDMA signals: Auxiliary-vector receivers," *IEEE Trans. Communications*, to appear.

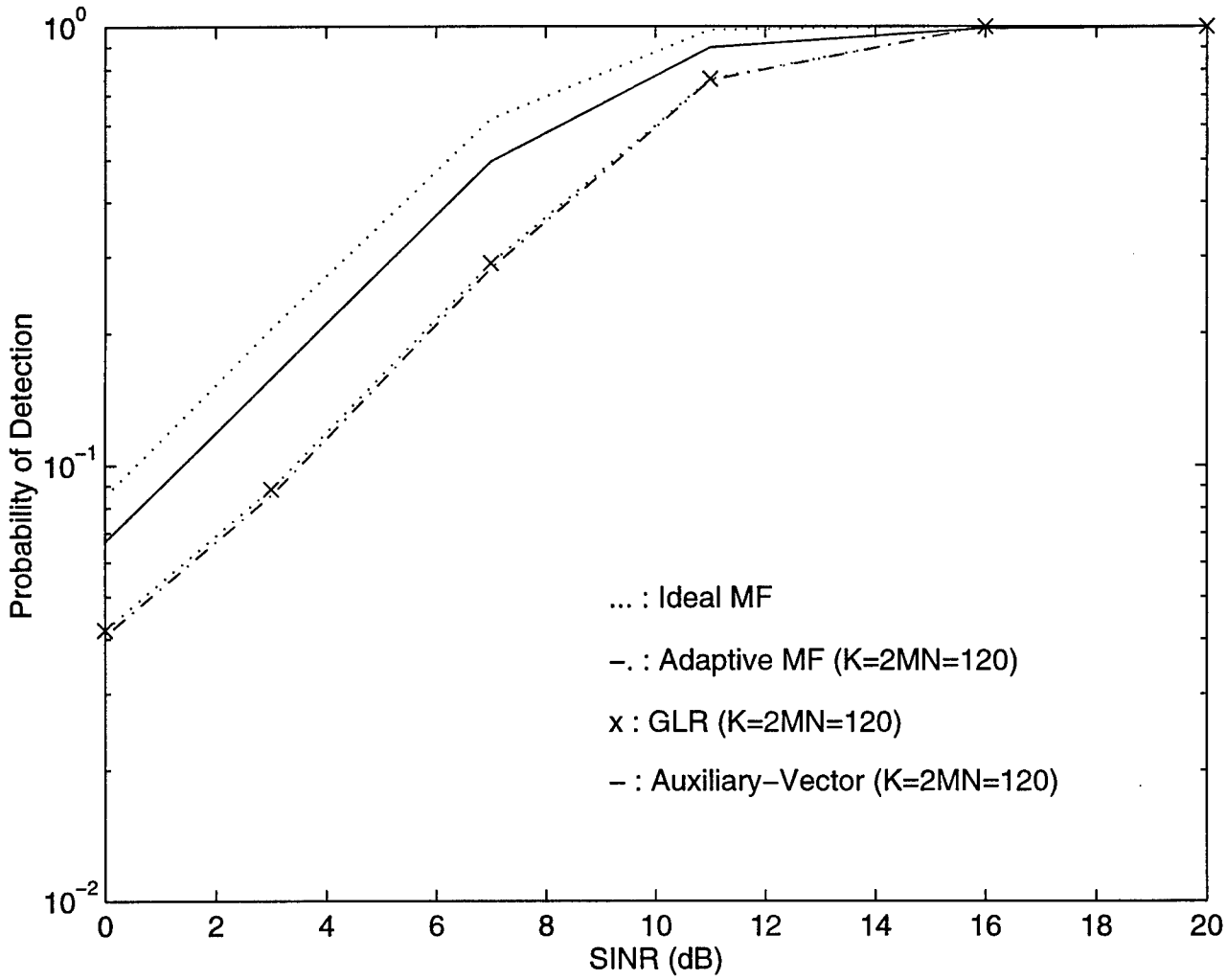


Figure 1: P_D versus SINR for $P_{FA} = 0.01$. The signal model includes three broadband interferers at 35dB and colored Gaussian noise (clutter) with covariance matrix R_{c_1} and "peak" CNR=20dB.

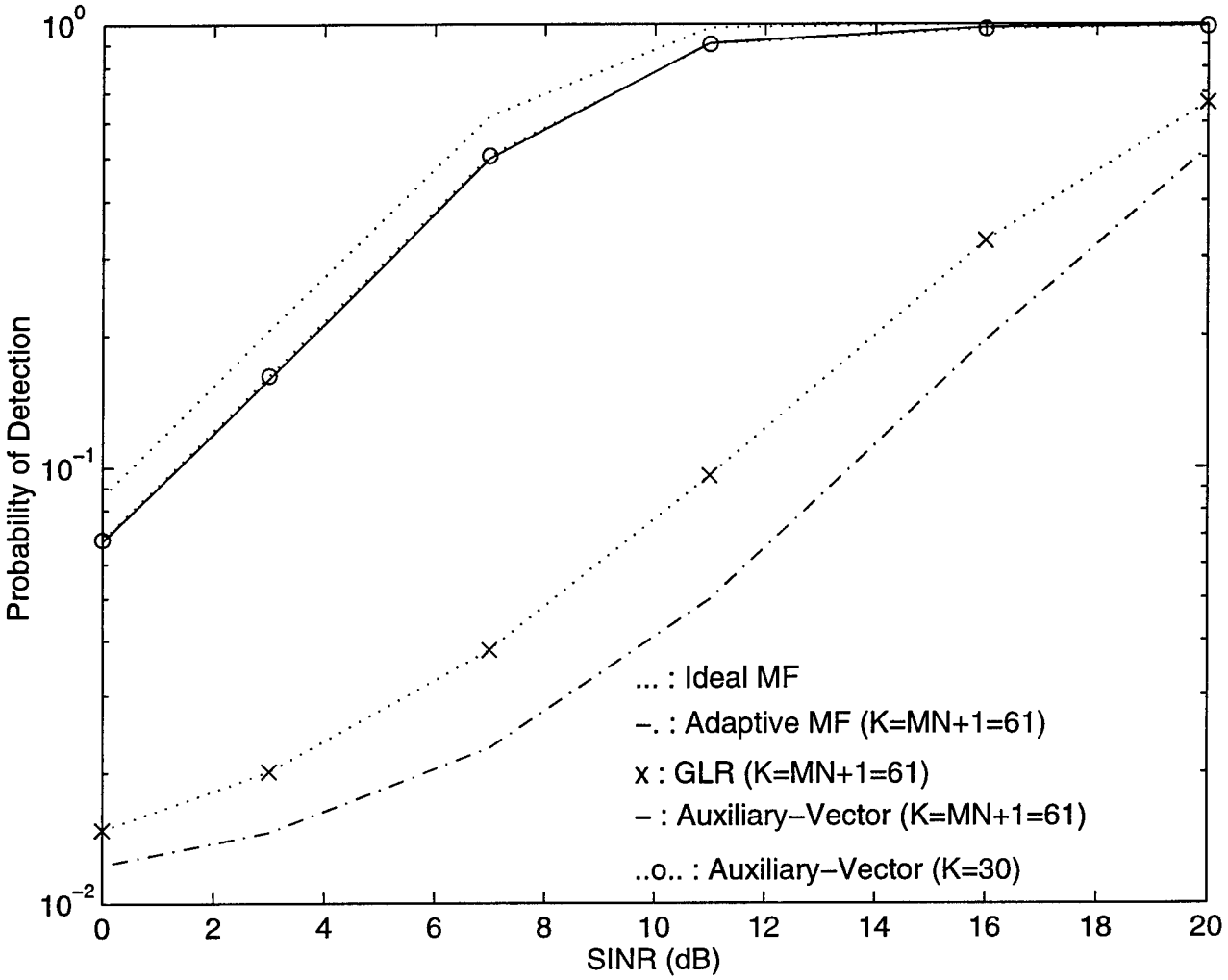


Figure 2: P_D versus SINR for $P_{FA} = 0.01$ and the same signal model as Fig. 1 (lower size secondary data sets are assumed).

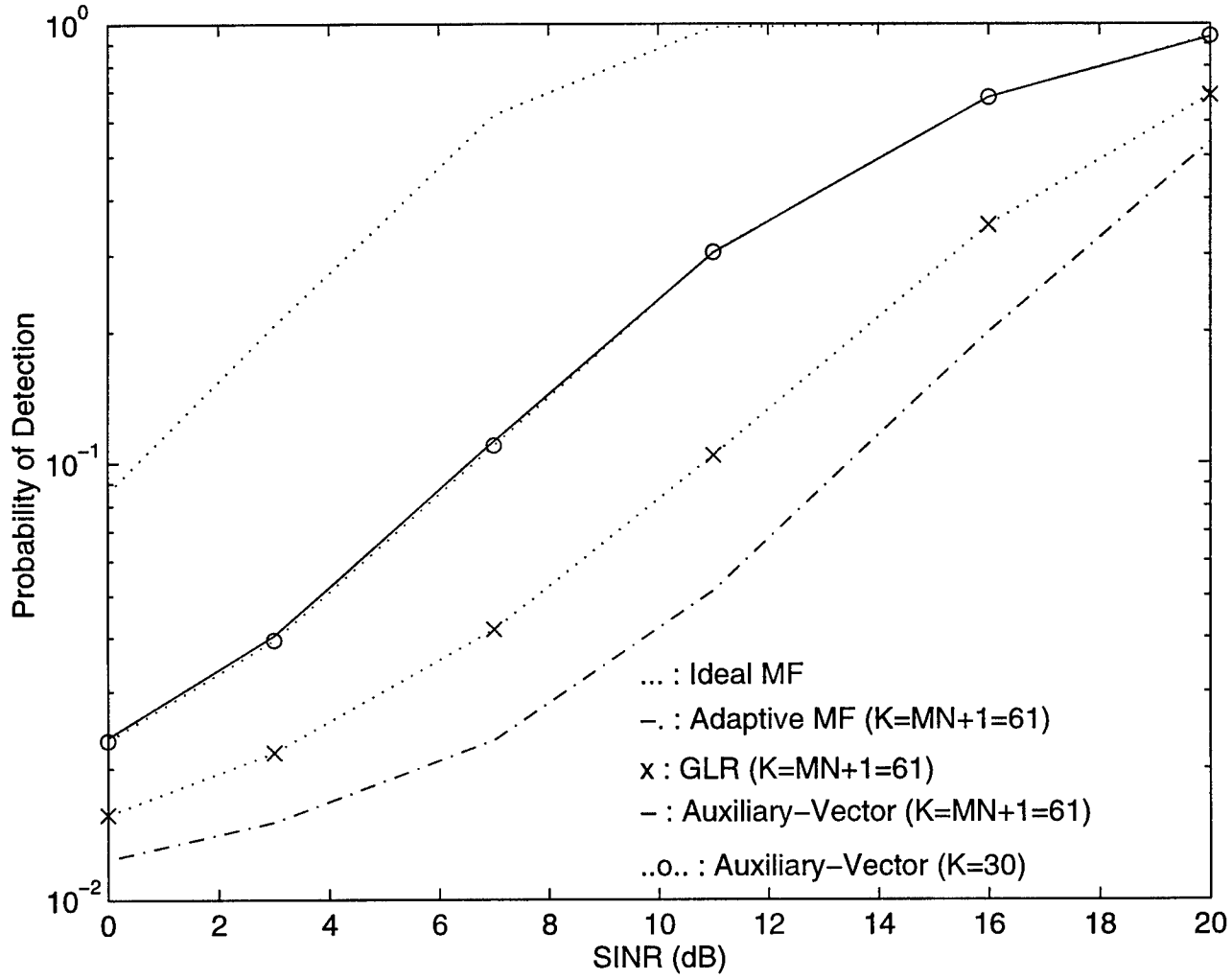


Figure 3: P_D versus SINR for $P_{FA} = 0.01$ and ill-conditioned colored Gaussian noise covariance matrix R_{c_2}.

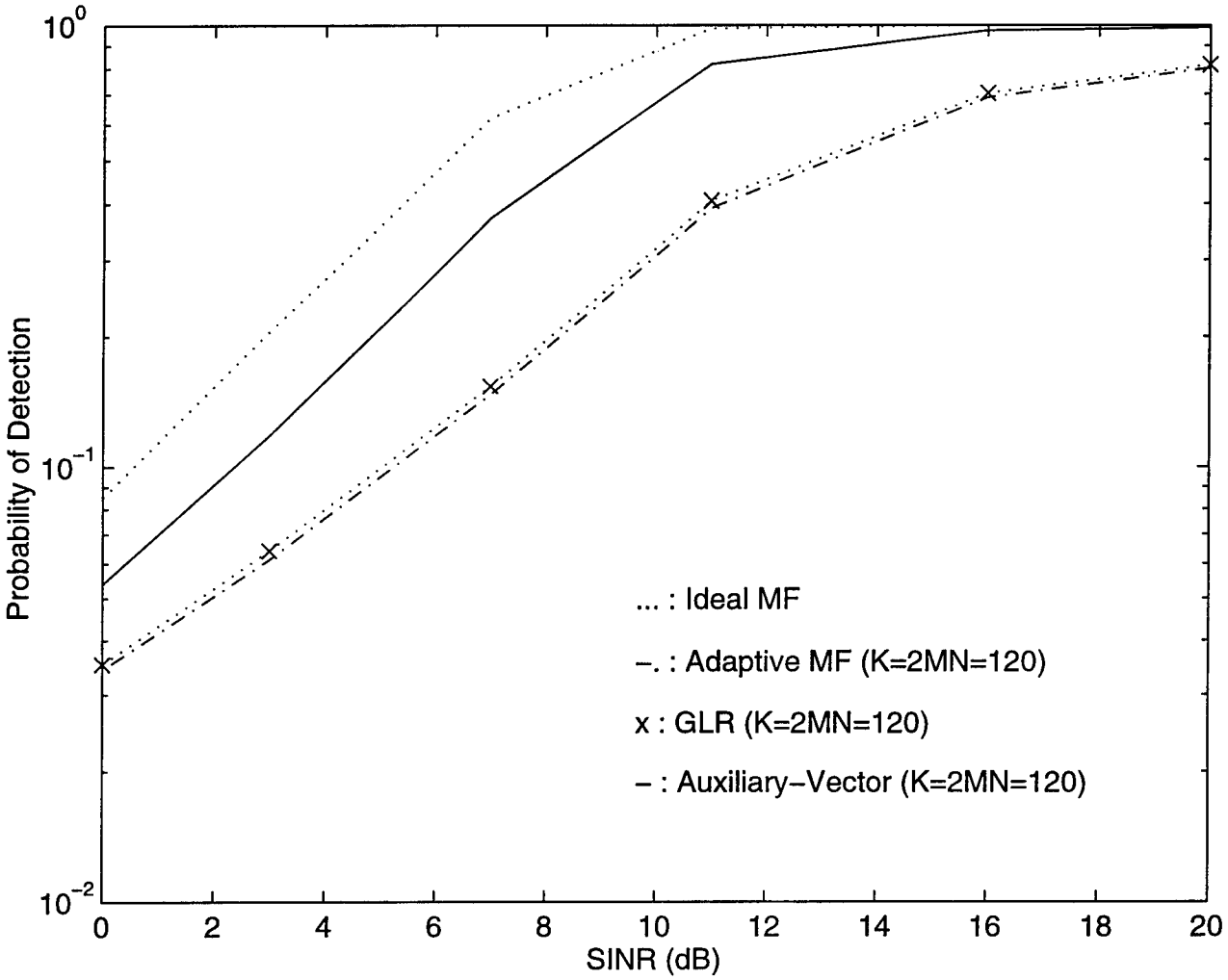


Figure 4: System and signal model as in Fig. 1. Secondary data set includes 12 samples from H_1.

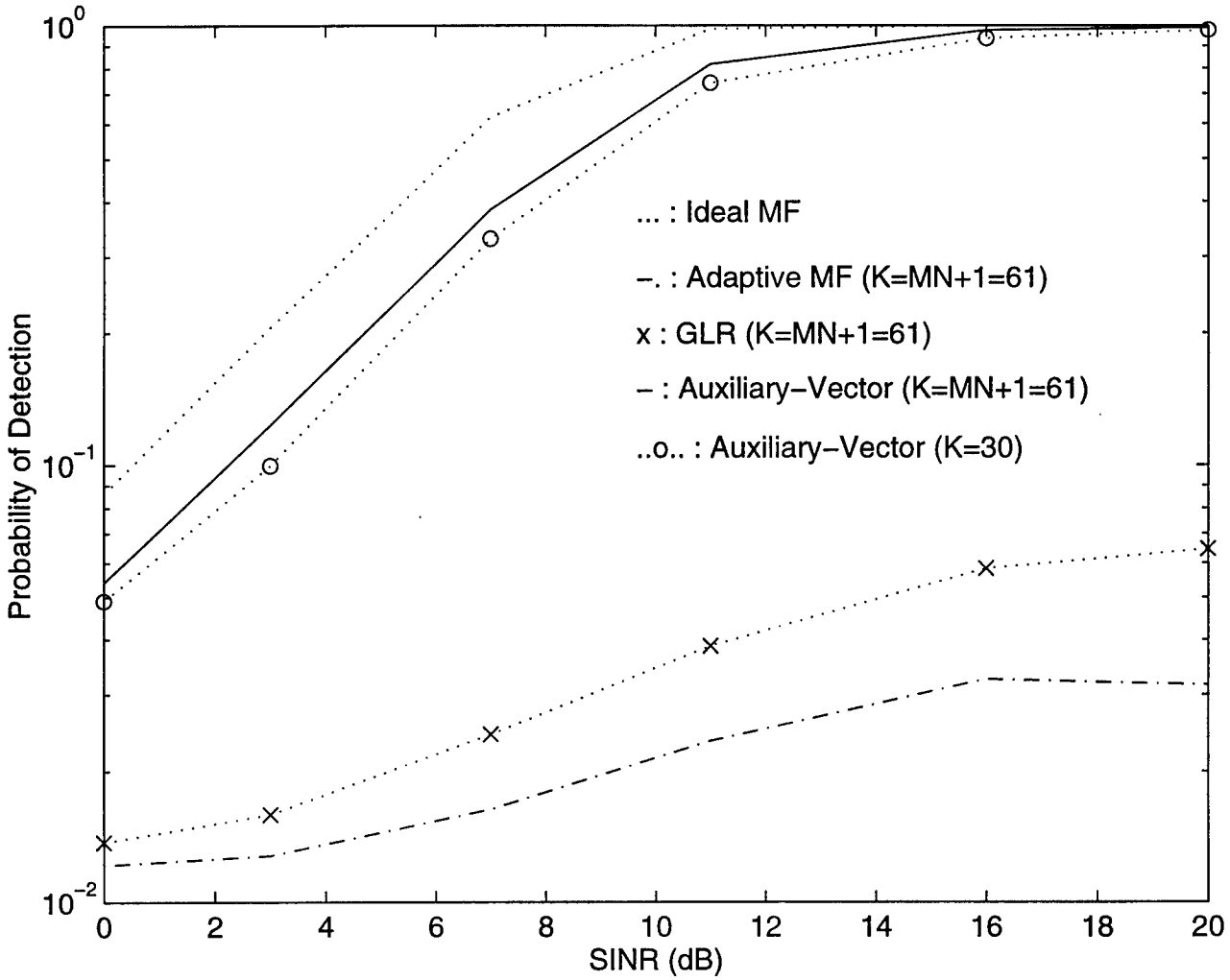


Figure 5: System and signal model as in Fig. 2. Secondary data set includes 6 out of 61, or 3 out of 30 samples from H_1.

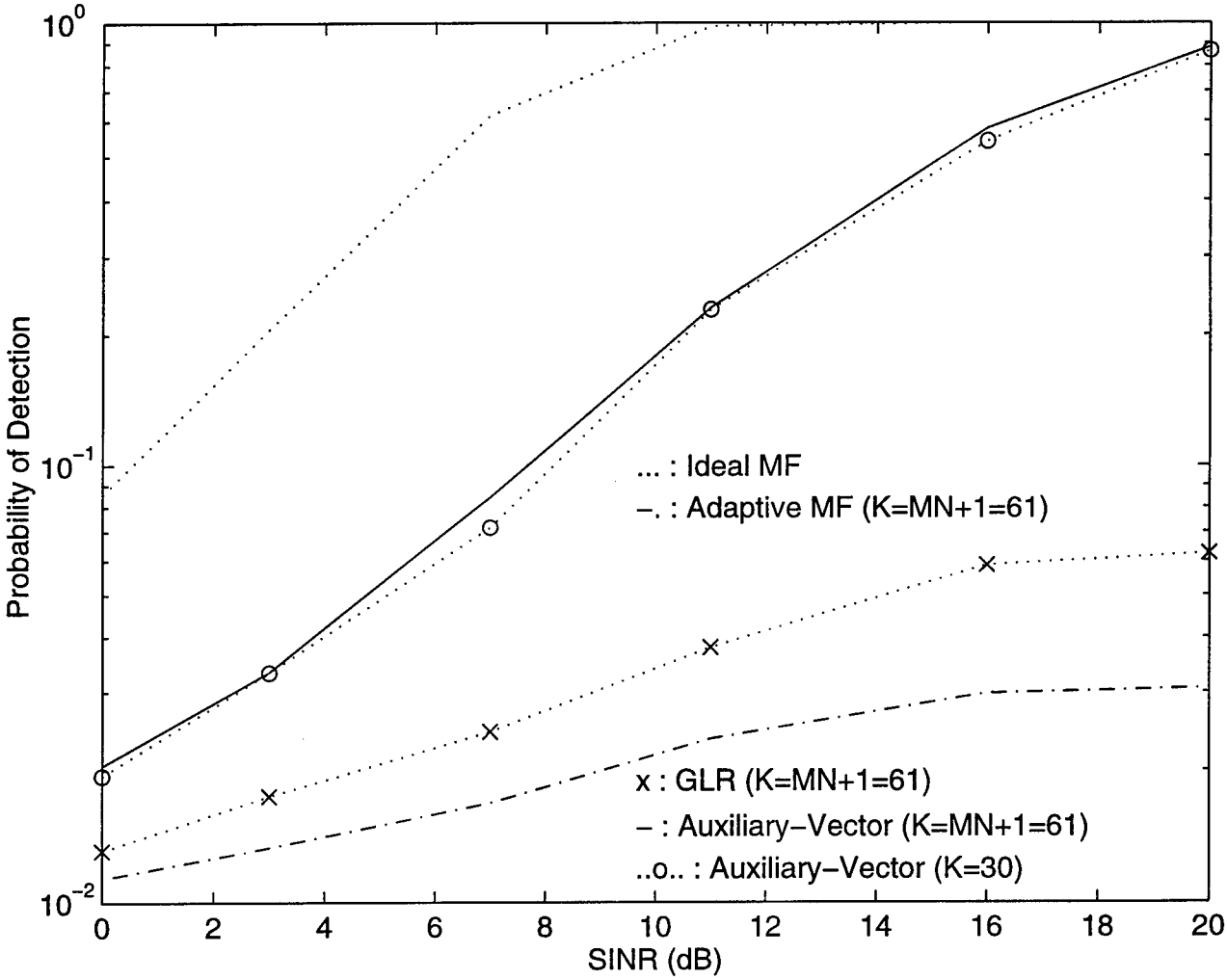


Figure 6: System and signal model as in Fig. 3. Secondary data sets as in Fig. 5.

A MODEL TO ATTAIN DATA INTEGRITY AFTER SYSTEM INVASION

Brajendra Panda
Associate Professor
Department of Computer Science

University of North Dakota
P.O. Box 9015
Grand Forks, ND 58202-9015

Final Report for:
Summer Research Program
Rome Laboratory

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, Washington, DC

And

Rome Laboratory

August 1997

A Model to Attain Data Integrity After System Invasion

Brajendra Panda
Assistant Professor
Department of Computer Science
University of North Dakota

Abstract

The objective of defensive information warfare is not only to prevent the system from malicious attackers, but also to detect such attacks when they occur and then to recover the system from the damage made by the attackers. This research is based on the assumption that the system under consideration has been attacked and the attack has been detected. A technique called data dependency has been developed that precisely determines the damage and recovers the system to a consistent state. The model achieves significantly improved efficiency over traditional transaction dependency approach.

A Model to Attain Data Integrity After System Invasion

Brajendra Panda

1. Introduction

The productivity of any organization heavily depends on the information it shares with and protects from the rest of the world. An attack through electronic media on an organization's information resources can have a significant impact on the organization's ability to operate. A mild attack may cause momentary disorder in the organizational activities where as a more prudent and well-planned attack can engender complete destruction of the information resources. Although there are several techniques available to prevent unauthorized access to sensitive data, these preventive measures are not always successful. It seems extremely difficult, although not impossible, to build systems that share information over electronic networks and still remain invulnerable to attackers. Hackers are always in search of new ways to prevail over the system security. Password sniffing and session hijackings are among various other means of intruding into a system, and the system will not be able to detect an attacker from a legitimate user in these cases. Besides, there remains possibility of significant damage by insider-turn-foes.

Defensive information warfare prepares the system to withstand such attacks and still provide system integrity. Three major aspects of defensive information warfare are prevention, detection, and recovery. A lot of research has been performed including strict access control mechanisms to protect the system from unauthorized users. Since prevention issues are outside the scope of this research, we would not reference them here. Ammann et. al. [Amma97] and Graubart et. al [Grau96] have discussed some of these issues with regard to information warfare environment. This research focuses on the recovery methods and assumes that an attack has been detected. Although not all attacks can be detected, there are numerous intrusion detection protocols that successfully watch suspicious activities of users - both insiders and outsiders - and detect any misuse. McDermott and Goldschalg have developed a method called storage jamming, [Mcd96a] and [Mcd96b], that uses fake data to attract attackers to update the values. These data are not used by regular users and their values are predetermined. Any change to these values confirms the activity of an intruder. Likewise, audit trails can be established to hold authorized users accountable for their activities and identify any misuse of privileges. A statistical approach to intrusion detection has been discussed in [Lunt90]. A survey of various intrusion detection techniques is discussed by Lunt in [Lunt93].

As discussed earlier, in most cases an attacker enters the system and accesses data as any other valid user. Thus, any update performed by the attacking transaction is made permanent and is available to other users to be read. Until the attack is detected, the damaged spreads through other users as they update other good data based on the damaged value. Therefore, timely recovery of the system is crucial to stop the cascading effect of the damage. Traditional recovery methods [Bers97], [Elma94], [Gray93], [Kort91] fail to provide the integrity and efficacy needed to react to the situation under consideration. These issues are discussed in the next section. The objective of this research is to make an exact assessment of the damaged data when the attack is detected and then recover the affected data in real-time.

In section 2, we examine existing recovery methods and their shortcomings in the information warfare environment. Section 3 discusses our recovery model. A graph based approach to the damage assessment is introduced in section 4. The algorithms are presented in section 5 and the advantages of this model are explained in section 6. The conclusion of this research and future research goals are discussed in section 7.

2. Related Work

Conventional recovery algorithms [Bers97], [Elma94], [Gray93], [Kort91] use a log to register each write operation of a transaction. During a system failure, the effects of all write operations of non-committed transactions are undone if they are already recorded in the stable database. Moreover, the effects of all write operations of committed transactions are redone if they are not in the stable database. This approach guarantees the integrity and consistency of the database. The log is also temporarily purged when it is determined that the stable database reflects the updates of all committed transactions (thus requiring no redo) and no effects of any of the non-committed transactions (thus requiring no undo). The recovery method does not require any read operations for any of the transactions. The transactions are also never stored entirely as only the before images and after images are required during the undo and redo processes.

The above approach does not work when there is a malicious transaction that has already updated a few data items and has committed. The system treats the attacker as any other valid transaction and makes the update permanent. This is guaranteed by the ACID properties (Atomicity, Consistency preservation, Isolation, and Durability) of transactions. Whenever the attacker is detected, all the updates of the attacker must be undone including the updates of the transactions that directly or transitively read from

the attacker. Then, these valid transactions must be re-executed to return the database to a consistent state. However, this is not possible for the following two reasons. First, as the log does not store the read operations, the read-from relationships can not be determined. Secondly, since the transactions are never stored entirely, redoing the valid transactions is impossible. Therefore, it is necessary to update the log to store both read and write operations, and also store the complete transactions in a separate log. Never the less, it is not efficient to re-execute all transactions from the point of attack. For example, if an attack is detected after a month of its occurrence, it requires significant amount of time to undo and then redo all the transactions that have directly or indirectly read-from the attacking transaction. Besides, the system remains unavailable to users during the recovery process and thus yields to denial of service. In most military and some commercial applications, denial of service is highly undesirable. Therefore, developing an efficient algorithm to recover the system from malicious attacks is quite essential.

3. Recovery Model

In this section we present the model on which our damage assessment and recovery algorithms are based. A requirement of our model is that in addition to the write operations, the log also stores the read operations. Moreover, we also assume that the scheduler produces a strict serializable history, and that the order of operations in the log is the same as that in the history. Next, we define data dependency, a condition on which this work is based.

Definition 1: A write operation $w_i[x]$ of a transaction T_i is *dependent* on a read operation $r_i[y]$ of T_i if $w_i[x]$ can not be performed without $r_i[y]$.

Definition 2: A data value *v1* is *dependent* on another data value *v2* if the write operation that wrote *v1* was dependent on a read operation on *v2*.

Definition 3: A write operation is called a *valid-write* if the value is written by a benign transaction and is independent of any contaminated data.

Note: A valid-write on a damaged data always refreshes the data bringing it to a consistent state.

Consider the following SQL statements:

1) UPDATE table1 SET *a1* = *a1* *1.02, *a2* = *a2* +5, *a3* = '*xyz*'
   WHERE *a2* > 50 AND *a4* is NOT NULL;

Let $v(a1)_t$ represent a value in attribute $a1$ at time t. According to the definitions,

$w[v(a1)_t]$ depends on $\{r[v(a1)_{t-1}], r[v(a2)_{t-1}], r[v(a4)_{t-1}]\}$,

$w[v(a2)_t]$ depends on $\{ r[v(a2)_{t-1}], r[v(a4)_{t-1}]\}$, and

$w[v(a3)_t]$ depends on $\{ r[v(a2)_{t-1}], r[v(a4)_{t-1}]]\}$.

2) UPDATE table1 SET $a1 = a1 * 1.02$, $a2 = a1 + 2.5$, $a3 =$ 'xyz';

In this case,

$w[v(a1)_t]$ depends on $r[v(a1)_{t-1}]$,

$w[v(a2)_t]$ depends on $\{ r[v(a2)_{t-1}]$, and

$w[v(a3)_t]$ is *independent* of all data values and guaranteed to be a valid-write.

It is neither always possible nor efficient to determine the dependency of operations as defined above due to the fact that the semantics of transactions are involved in this process. A more plain and effective definition that is adopted in this research is given below.

Definition 4: A write operation $w_i[x]$ of a transaction T_i is *dependent* on a read operation $r_i[y]$ of T_i if $r_i[y]$ is scheduled before $w_i[x]$ in the log.

That is, a write operation of a transaction depends on all read operations of the same transaction that precede the write operation. Hence, the value dependency can be determined accordingly.

Consider the history $H = r_1[a]\ r_1[b]\ w_1[c]\ r_2[a]\ w_2[b]\ w_2[d]\ c_2\ r_3[d]\ r_3[a]\ c_1\ r_3[c]\ w_3[d]\ c_3\ r_4[b]\ c_4\ w_5[a]\ w_5[b]\ c_5\ r_6[b]\ w_6[b]\ r_6[c]\ w_6[c]\ r_6[d]\ w_6[d]\ r_6[a]\ w_6[a]\ c_6$.

Note that the operation c_i represents the commit operation of transaction i. This is the final operation that is executed, after T_i completes successfully, to make the effects of T_i permanent in the stable database. As per definition 4, $w_1[c]$ in history H is dependent on $r_1[a]$ and $r_1[b]$. For that reason, the value of c written by T_1 is dependent on the values of a and b. The dependencies of other operations in H can be derived accordingly.

Notation: Let $p = T_n(q_1, q_2, ..., q_k)$ denote that the value of p written by T_n depends on values of q_1, q_2, ..., and q_k read by T_n.

Scanning the history H, we get the following dependency of values:

$c = T_1(a,b)$, $b = T_2(a)$, $d = T_2(a)$, $d = T_3(a,c,d)$, $a = T_5()$, $b = T_5()$, $b = T_6(b)$, $c = T_6(b,c)$,

$d = T_6(b,c,d)$, and $a = T_6(a,b,c,d)$.

Table 1 shows the above data dependencies in H as they occurred over time. Let us assume that T_2 has been determined as the attacker. Therefore, items b and d are concluded to be damaged. Later, when T_3 writes d after reading it, d continues to be damaged. As the value of b written by T_5 is independent of any contaminated data, b has been refreshed. T_6 has read and updated b and c prior to reading the damaged data d. Therefore, b and c remain fresh. Likewise, d still remains damaged and, furthermore, a is damaged as $w_6[a]$ is dependent on $r_6[a]$.

a	b	c	d
		$T_1(a,b)$	
	$T_2(a)$		$T_2(a)$
			$T3(a,c,d)$
$T_5()$	$T_5()$		
	$T_6(b)$		
		$T_6(b,c)$	
			$T_6(b,c,d)$
$T_6(a,b,c,d)$			

Table 1: Data Dependency for History H.

4. Damage Assessment

A graph based approach has been used in this section to observe how damage has spread through the database and which damaged items are refreshed via valid-writes. The graph we use for this purpose is a directed graph. A node in the graph represents a new value of a data item at a given time, and contains information such as the data item name, time of update, and a boolean value indicating whether the data is contaminated or not. For simplicity, we symbolize each node by either a circle or by a square having the data item name inside it. A circle denotes a clean data item while a corrupted data item is denoted by a square. Each edge represents an update by a transaction that either corrupted the data, or transmitted the damage, or refreshed the data. The nodes for any particular data item are drawn vertically below one another to specify the order among them with respect to the time of update.

The graph starts with all circular nodes, one for each data item in the database. Whenever an attack is detected, a square-type node is created for each data item that is updated by the attacker. An edge is

added to each new node from the initial node that represents the same data item. The edge carries the identification of the updating transaction. For each update in the log that depends on the damaged data, a square-type node is created and an edge is added to the new node from the node(s) on which the update depends. Moreover, whenever a transaction performs a valid-write operation on a damaged data item, a circular node is created for the item and an edge from the previous square-type node of the same item is added showing that the damage has been repaired. Thus, we have three types of edges in the graph: the edges that denote the initial corruption of data by attacking transactions, the edges that show transmission of damage, and the edges that indicate the refinement of damaged data through valid-writes. Note that no other valid-writes appear in the graph except for those representing the edges of the third type. Figure 1 shows the graph built from history H. The graph depicts that items b and c were originally damaged by transaction T_2 (attacker). Later b has been restored through a valid-write by T_5. However T_3 and T_6 carried on the damage on d. Moreover, T_6 also contaminated a by updating it after reading d.

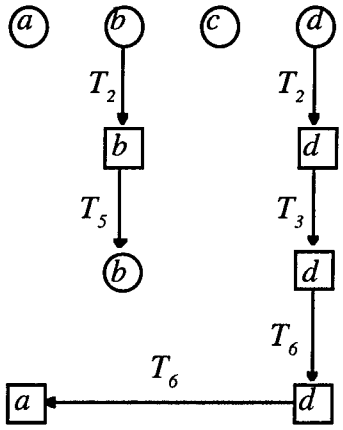


Figure 1: Damage Assessment in H using a Directed Graph

Of all the data dependency derived from H, only the following contribute to the spread and recovery of the damage. Also observe that exactly these dependencies are represented by the edges in the graph.

$b = T_2(a)$, $d = T_2(a)$, $d = T_3(a,c,d)$, $b = T_5()$, $d = T_6(b,c,d)$, and $a = T_6(a,b,c,d)$.

Figure 2 displays a more complex graph that is obtained as an extension of the graph in figure 1. These extensions are based upon additional database transactions. The discontinuity in the transaction number is to designate the missing transactions as the benign transactions that do not contribute to spread of the damage and hence do not participate in the graph. The dependencies considered for the construction of the graph in figure 2 are:

$b = T_2(a)$, $d = T_2(a)$, $d = T_3(a,c,d)$, $b = T_5()$, $d = T_6(b,c,d)$, $a = T_6(a,b,c,d)$, $b = T_{10}(a, d)$, $a = T_{12}(a,c)$,

$d = T_7(c)$, $a = T_{14}(c)$, $c = T_{11}(b)$, and $c = T_{17}(a,d)$.

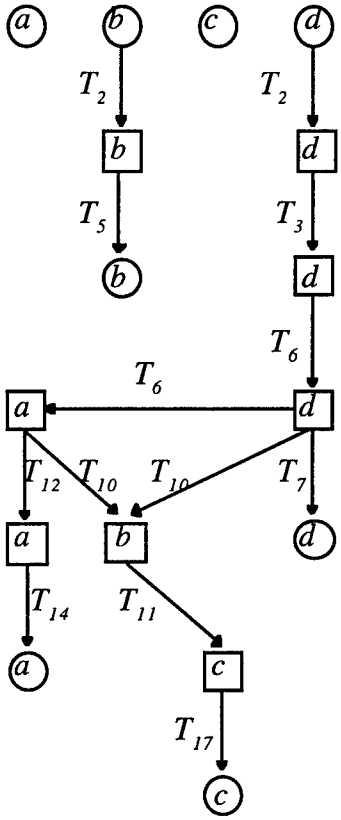


Figure 2: A More Complex Graph on Damage Assessment

Note that at the end, only item b is detected as damaged, whereas items a,c, and d have fair values. Although b was initially contaminated by T_2, during recovery we do not need to undo the effect of T_2 in order to recover b. This situation arises from the fact that edges from the node of b that was contaminated by T_2 do not lead to the final contaminated state of b. To recover the correct value of a, however, we need to trace back all the way to the value of d just before T_2, the attacker, updated it. Since d has a fresh value at the time of recovery, the old value of d should only be calculated to compute the value of a and b rather than updating d in the database. Also note that, although T_2 had updated b and d, we need not compute the result of the operation of T_2 on b. That is, only the operations representing the edges that lead to the current damaged data needs to be recomputed.

5. The Algorithms

We present two algorithms here. The first algorithm performs damage assessment and recovery simultaneously; whereas the second algorithm separates damage assessment and recovery processes for

improved efficiency. Due to space constraints, the proofs of correctness for these algorithms are not provided in this paper. Interested readers may contact the author. As stated earlier, we require that the log be modified to store the read operations of every transaction. This modification is necessary to determine the dependency of operations. Although the value read can be determined from the after image of the previous write operation on the same data item, for optimization reasons, the value read may as well be stored along with the read operation. In the following algorithm, fresh_list and read_list_T_i are lists of records with two fields: data item field and value field. The structure damage_item_list includes the list of damaged data as concluded by the algorithm. The value of each damaged item in the damaged_item_list that would have been in the database if the attacking transaction had not been executed, is calculated and stored in the fresh_list along with its associated data item field. The read_list_T_i contains the data items and values that are read by T_i.

Notation: Let $[T_i, x, v1, v2]$ denotes the write operation of T_i in the log where $v1$ and $v2$ are respectively the before and after images of x. The read operations of T_i are denoted by $[T_i, x, v]$ which indicates that the value of x read by T_i is v.

This notation of representing a write operation of a transaction has been used in [Elma94]. We present our first algorithm on damage assessment and recovery below. This algorithm assesses the damage and calculates the fresh value of the damaged data simultaneously. Note that, as mentioned earlier, the algorithm is based on the assumption that one or more transactions have been identified as attackers. Perhaps the intrusion detection mechanism, or an human analyst checking the values of known data items in the database, or some other means helped in the identification process.

Algorithm 1

1. Set damage_item_list = { }; /* Empty set */
2. Set fresh_list = { };
3. Scan the log until the end
   - 3.1 For every write operation $[T_i, x, v1, v2]$ of an attacker, if $x \notin$ damage_item_list,
     add x to damage_item_list; and
     add the record $(x, v1)$ to fresh_list; /* $v1$ is the before image of x */
   - 3.2 For any other transaction T_j appearing after a write operation of the first attacker,
     set read_list_T_j = { };
   - 3.3 For every read operation $[T_j, x, v]$ of T_j,

add record (x, v) to read_list_T_j;

3.3.1 If $x \in$ damage_item_list,

replace value v of record (x, v) in read_list_T_j by value of x in fresh_list;

3.4 For every write operation $[T_j, x, v1, v2]$ of T_j,

3.4.1 If the set of data items in read_list_$T_j \cap$ damage_item_list $\neq \phi$,

recalculate new value $v2$ of x, by using values in read_list_T_j;

3.4.1.1 If $x \in$ damage_item_list,

replace value of x in fresh_list by new value $v2$;

3.4.1.2 Else

add record $(x, v2)$ to fresh_list; and

add x to damage_item_list;

3.4.2 Else

3.4.2.1 If $x \in$ damage_item_list,

remove x from damage_item_list;

remove the record of x from the fresh_list;

4. For each item in damage_item_list,

replace its value in the database by the value in the fresh_list.

Once the list of contaminated data is determined, all data items in the list are blocked from being read by other transactions. This will stop further spread of the damage in the database. However, while recovering the damaged data, overwrites on them by any active transaction must be allowed. Such an overwrite will be a valid-write because no damaged data is allowed to be read. This option will refresh the damaged values. Once a damaged data is refreshed through the recovery process or through a valid-write, the data can be made available for read/write purposes.

Some of the steps in the above algorithm require explanation. Since the attacking transaction may not be the first one of this type, there is a possibility that the data item updated by this particular transaction may have been damaged through a previous attacker. Therefore, in step 3.1, the damage_item_list is checked to see if x is already there. In that case, neither the damage_item_list nor the fresh_list need to be updated. Since the fresh_list should have a correct value of x, that value is the before image for this update and must be left there. However, if x was not damaged previously or was damaged but has been refreshed, then the previous image of x is the correct value of x. The value $v1$ in item $[T_j, x, v1, v2]$ of the log is the previous image and, therefore is inserted into the fresh_list. When T_j reads a damaged data x,

in step 3.3.1, the correct value of x that is stored in the fresh_list (see theorem 2 below) is appended to the read_list_T_j for correct calculations of any future updates made by T_j. In step 3.4.1, *v2* is the correct value that should have been in the database if the attacking transaction was not executed. Step 3.4.2.1 will be executed only when the transaction has not read any damaged data but blind-wrote a damaged data. This operation refreshes the damaged value of the data. So the data item is removed from the damage_item_list and from the fresh_list.

Although the previous algorithm will precisely detect all damaged data items and repair them, it will block all the active transactions in the system until the recovery process is complete. During this process a significant delay is expected due to the computation required to determine the valid values of all damaged data and the disk access needed to the log that keeps a copy of all committed transactions in the system. The next algorithm removes these problems by first determining the set of damaged data and then making non-damaged data available to active transactions in the system. This will enable the unaffected part of the database be operative while the recovery continues. As pointed out earlier, during the recovery process the damaged data are available for blind-writes. This further increases the availability of data in the system while the recovery is in progress.

It is possible to accomplish the above mentioned goal by removing the fresh_list and the related calculation from the previous algorithm during the damage assessment phase. Then, during recovery, the new value of each damaged data can be calculated. In order to do this, however, the partial order of all transactions that have accessed damaged data is needed. The process involves a second reading of the log incurring more disk accesses. To resolve this problem, we use a different data structure, damage_audit_table, as described next.

The auxiliary structure damage_audit_table stores information about transactions that have either corrupted the data, or transmitted the damage, or refreshed the data. Transaction records are appended to the list in the same order as they appear in the log. Each record has four fields: transaction_id, data_written, valid_read, and invalid_read. The data_written field stores only data items with values that are either written by an attacker, or the data that are dependent on damaged data, or the fresh value of a previously damaged data after a valid-write. Notice that all the data items in this column also appear as nodes in the damage assessment graph and vice versa. While the valid_read field stores all non_damaged data with their values as read, the invalid_read field keeps the damaged data along with their values that are read by the transaction. The algorithm consists of two phases: the damage assessment phase and the

recovery phase. We present the damage assessment phase next. Again, note that the following algorithm requires the identification of the attacking transaction(s) as assumed for the previous algorithm.

Algorithm 2.1 (Damage Assessment)

1. Initialize damage_audit_table = { }; and damage_item_list = { };
2. Scan the log until the end
   - 2.1 When an attacker, T_i, is found,
     add a new record with the transaction_id of T_i into damage_audit_table;
     - 2.1.1 For every write operation $[T_i, x, v1, v2]$ in the log,
       add $(x, v1)$ order pair to data_written column of T_i's record; and
       add x to the damage_item_list if it is not there;
   - 2.2 For any other updating transaction T_j appearing after a write operation of the first attacker, add a record for T_j into damage_audit_table;
     - 2.2.1 For every read operation $[T_j, x, v]$
       - 2.2.1.1 If $x \in$ damage_item_list, add x to invalid_read column of T_j;
         Else add (x, v) pair to valid_read column of T_j;
     - 2.2.2 For every write operation $[T_j, x, v1, v2]$
       - 2.2.2.1 If invalid_read column of T_j is not empty, /* T_j has spread the damage */
         add $(x, v2)$ pair to data_written column of T_j; and
         add x to the damage_item_list if it is not there;
       - 2.2.2.2 Else
         - 2.2.2.2.1 If $x \in$ damage_item_list, /* T_j has a valid-write on damaged data */
           write $(x, v2)$ pair to data_written column of T_j; and
           remove x from damage_item_list;
         - 2.2.2.2.2 Else /* T_j neither read not updated damaged data */
           remove T_j's record from damage_audit_table.

Once the damage_list is determined, all non-damaged data are made accessible to users while the recovery process continues. However, users are allowed to make blind-updates on damaged data although they would not be able to read such data. Next, we present the recovery phase of the algorithm. This algorithm uses the damage_audit_table and the damage_item_list as the input in determining the correct value of the damaged data.

Algorithm 2.2 (Recovery)

1. Scan the records in damage_audit_table until the end;
   1.1. For every attacking transaction other than the first one,
      1.1.1 For each (x, v) pair in data_written column
         substitute v by the before image of x;
      /* For the first attacker, the previous image is already there */
      /* If multiple attackers have written x consecutively, the before image of x for each of these transactions is that of the first transaction */
   1.2. For every non-attacking transaction with non-empty invalid_read column
      /* These transactions have spread the damage */
      /* Any non-attacking transaction with an empty invalid_read column in damage_audit table, has refreshed some data items and their records need not be modified */
      1.2.1 For every x in invalid_read column,
         scan the data_written column upward starting from the previous record in damage_audit table to find the first (x, v) pair; /* the last update on x */
         add (x, v) pair into the valid_read column; and
         remove x from invalid_read column;
      1.2.2 For every x in data_written column,
         calculate the value v of x using values in the valid_read column; and
         substitute (x, v) in data_written column; /* v is the correct value of x */
2. For every x in damage_item_list, check the new log that has just been created while the recovery process was in progress;
   2.1 If x is not modified in the log, /* otherwise, the update on x is a valid-write */
      scan data_written column of damage_audit table from bottom-up to find first (x, v) pair, then substitute the value of x in the database with v.

Table 2 displays the damage_audit_table that would be constructed during the damage assessment algorithm (algorithm 2.1) if the history containing data dependencies for graph 2 would be used. In the figure, the notation x_b and x_a represent the before and after images of x respectively.

6. Advantages of this Model

Our model is based on the data dependency approach as opposed to the traditional transaction dependency approach. In the latter case, a transaction T_i is said to be dependent on T_j if T_i has read from T_j. The read-from relationship has been discussed in [Bers97]. In the transaction dependency method, if a transaction has read damaged data, all updates made by that transaction, and all updates made by any subsequent transactions that depend on the prior ones are concluded to be damaged. However, in data dependency approach, only those data items that have been modified based on the value of any damaged data are consider damaged. In the following paragraphs we address few of the advantages of using our new approach over the conventional transaction dependency approach.

T_Id	Data_written	Valid_read	Invalid_read
T_2	(b, b_b), (d, d_b)		
T_3	(d, d_a)	(a, v), (c, v)	d
T_5	(b, b_a)		
T_6	(d, d_a), (a, a_a)	(b, v), (c, v)	d
T_{10}	(b, b_a)		a, d
T_{12}	(a, a_a)	(c, v)	a
T_7	(d, d_a)	(c, v)	
T_{14}	(a, a_a)	(c, v)	
T_{11}	(c, c_a)		b
T_{17}	(c, c_a)	(a, v), (d, v)	

Table 2: Damage_audit Table Based on History of Graph 2

By evaluating the damage through data dependency, precise information on damaged data is obtained. The transaction dependency method would falsely mark data that are updated prior to reading any corrupted data as damaged. Any transaction performing updates that depend on these falsely marked data will generate a cascading effect of spreading the amount of illusory damage. To illustrate the effect, consider that every transaction that is assumed to have read a damaged data has updated two other data items one of which is independent of the damaged item read. For simplicity, also assume that every such affected data item is accessed by only one transaction. To start with, consider that data item a is damaged. The first transaction updates b and c, and has read a. As per transaction dependency, it is

concluded that both *b* and *c* are contaminated. Figure 3 shows a tree representing damage propagation in transaction dependency model after six other transactions of the above mentioned type finish execution.

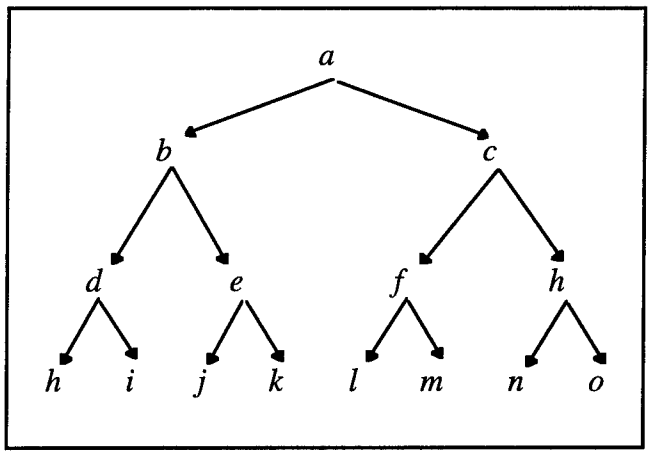


Figure 3: Illusory Damage

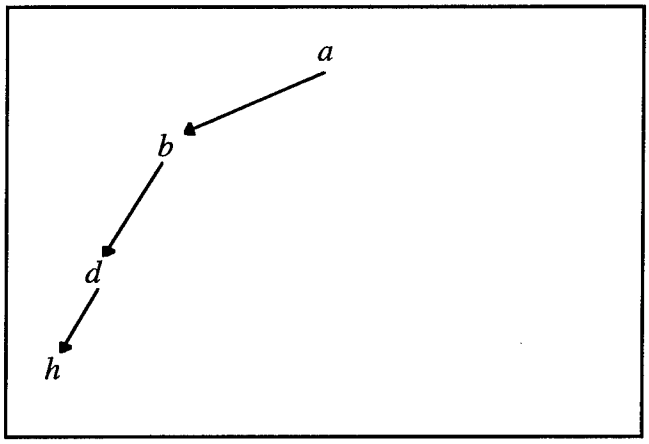


Figure 4: Actual Damage

But, in reality, only one item in each level of the tree in figure 3 is actually contaminated. Since out of the two updates for every transaction, only one item is dependent of the damaged data, for clarity purpose, assume that the left child of every node in the tree is damaged whereas the right child is not. Therefore, at the second level (considering item *a* at level 1)*b* is damaged and *c* is not. At the next level, only *d* is damaged and *e* is not. However, both *f* and *g* are not damaged since *c* is not damaged. Similarly, at the leaf level of the above tree, only *h* is damaged and the rest are not. Hence, the actual list f damaged data in this example is: *a*, *b*, *d*, and *h* (see figure 4).

As can be observed, at the second level, after completion of the first transaction, there is only one falsely marked data item out of 3 items in total, resulting in a 33% of false markings. At the next level, the percentage of falsely marked data is 57%, 4 out of 7. This number is 73% (11 out of 15) at the leaf level of the above tree in figure 4. As the level of the tree increases, the number of incorrectly marked data rises tremendously. In general, in a tree of level n, considering root as level 1, there will be exactly n damaged data items, (2^n) - $(n+1)$ falsely marked data items out of (2^n)-1 totally accessed data. Thereupon, the percent of falsely marked data to the total accessed data is $(((2^n) - (n+1)) * 100) / ((2^n)-1)$. Just to substantiate the significance of this number, consider a tree of level 8. There will be 96.8% of falsely marked data in this tree as per the transaction dependency method. Note that this is a very rough estimate as various other necessary parameters are not considered here.

Two major problems arise as a result of false markings of data. First, this will needlessly prolong the recovery process. Secondly, some transactions that could have accessed these incorrectly marked data otherwise, would be delayed until the recovery process is complete. This process will also generate a cascading effect since these blocked transactions may be holding some non-damaged data items while waiting for the recovery process to be over. This effect is substantially minimized by knowing exactly which data items are damaged and thus making rest of the database available to users.

Moreover, by allowing blind writes in our model, some damaged data are automatically recovered. Although blind-writes are not common for numeric data, most updates on non-numeric data (name, address, phone numbers, etc., for example) will be of this category as updates on these items normally do not depend on their previous values. In our model, the recovery is performed much quicker for the following reasons. First, the amount of corrupted data is precisely determined. Secondly, instead of re-executing all affected good transactions, our method executes only the damage dependent operations of these transactions. Furthermore, as the transactions are not executed entirely, there will be no contention among them either.

7. Conclusions

In an information warfare situation, the database may become corrupted through electronic attacks on the system. It may not be possible to detect such an attack immediately. The system would commit the attacking transaction as any other transaction and make the effect permanent. The damage would spread via other benign transactions whose updates depend directly or indirectly on the damaged data. The existing recovery algorithms are not designed to operate in an information warfare environment. This

research offers recovery algorithms that restore the database to a consistent state by recomputing the affected operations of all benign transactions that follow the attacker. For this purpose, the transaction log is modified to store the read operations of all transactions in addition to their write operations. The first algorithm performs the damage assessment and recovery simultaneously while these two methods are separated in the second algorithm. Though the first algorithm is quicker than the second, it requires that the entire system is brought to a halt until the recovery is complete. The second algorithm, as comprises two phases and therefore slower than the first, releases the unaffected part of the database soon after the assessment phase is completed. This makes the system available to users while the recovery process continues. We have discussed the advantages of using data dependency over transaction dependency and the tremendous efficiency obtained by this method.

Future research directions include the development of a quantitative analysis of this model to show the efficiency of the data dependency model compared to other, traditional recovery approaches. Besides, some other issues such as garbage collection to reduce the amount of log records, using prior knowledge of data dependency in the next recovery process, computing only operations that are required in the final update of damaged data, etc. needs to be investigated. Furthermore, we plan to consider possible implementation issues using a relational database management system.

Acknowledgments

The author sincerely acknowledges the support of Dr. Heather Dussault and Joe Giordano of Rome Laboratory during this research work. This work was made possible by the summer research program of AFOSR.

References

[Amma97] P. Ammann, S. Jajodia, C. D. McCollum, and B. Blaustein, "Surviving Information Warfare Attacks on Databases", In Proceedings of the 1997 IEEE Symposium on Security and Privacy, p.164-174, Oakland, CA, May 1997.

[Bers97] P. A. Bernstein, V. Hadzilacos, and N. Goodman, "Concurrency Control and Recovery in Database Systems", Addison-Wesley, Reading MA, 1987.

[Elma94] R. Elmasri and S. B. Navathe, "Fundamentals of Database Systems", Second Edition, Addison-Wesley, Menlo Park, CA, 1994.

[Gray93] J. Gray and A. Reuter, "Transaction Processing: Concepts and Techniques", Morgan Kaufmann, San Mateo, CA, 1993.

[Grau96] R. Graubart, L. Schlipper, and C. McCollum, "Defending Database Management Systems Against Information Warfare Attacks", Technical report, The MITRE Corporation, 1996.

[Kort91] H. F. Korth and A. Silberschatz, "Database System Concepts", Second Edition, McGraw-Hill, New York, 1991.

[Lunt90] T. F. Lunt, "Using Statistics to Track Intruders", In Proceedings of the Joint Statistical Meetings of the American Statistical Association, August 1990.

[Lunt93] T. F. Lunt, "A Survey of Intrusion Detection Techniques", Computers & Security, Vol. 12, No. 4, p. 405-418, June 1993.

[Mcd96a] J. McDermott and D. Goldschlag, "Storage Jamming", Database Security IX: Status and Prospects, D. Spooner, S. Demurjian, and J. Dobson, editors, p. 365-381, Chapman & Hall, London, 1996.

[Mcd96b] J. McDermott and D. Goldschlag, "Towards a Model of Storage Jamming", In Proceedings of the IEEE Computer Security Foundations Workshop, p. 176-185, Kenmare, Ireland, June 1996.

PHASE TRANSITIONS IN PROBABILITY ESTIMATION AND CONSTRAINT SATISFACTION PROBLEMS

Michael A Pittarelli
Associate Professor
Department of Computer Science

State University of New York
P.O. Box 3050
Utica, NY 13504-3050

Final Report for:
Summer Research Program
Rome Laboratory

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, Washington, DC

And

Rome Laboratory

August 1997

PHASE TRANSITIONS IN PROBABILITY ESTIMATION AND CONSTRAINT SATISFACTION PROBLEMS

Michael A. Pittarelli
Associate Professor
Department of Computer Science
SUNY Institute of Technology at Utica/Rome

Abstract

Experiments were conducted to identify phase transitions in two domains: estimation of multinomial probability distributions from observed relative frequencies, and determination of the consistency of database frequency tables. The database consistency problem required that the degree to which an instance is constrained (on the average) be represented in a unique way, as the superposition of two weighted undirected graphs.

PHASE TRANSITIONS IN PROBABILITY ESTIMATION AND CONSTRAINT SATISFACTION PROBLEMS

Michael A. Pittarelli

Two sets of experiments were planned and conducted. Experiments were conducted to test the performance of a probability estimation technique based on a search of a lattice of graphical models. Experiments were also conducted to search for hard instances of a particular NP-complete problem, that of determining the consistency of a partially specified set of database frequency tables. The results for the database consistency problem were especially interesting.

I. Multinomial Probability Estimation Experiments

A number of techniques exist for adjusting probabilities estimated as relative frequencies from small samples. The adjustment is typically in the direction of a more uniform distribution. The target distribution needn't be the uniform distribution, although it typically is. An alternative "ultrasmooth" distribution is uniform over all but specified events; for those events, the probability is zero (so-called "structural zeros").

A number of different classes of adjustment techniques are discussed by Titterington [1]. One important class is the class of *convex smoothing* methods, for which

$$\tilde{p} = \lambda p_N + (1 - \lambda)s,$$

where:

$\lambda \in [0, 1]$

p_N is a relative frequency distribution based on N observations

s is an ultrasmooth distribution (over the same set of events that p_N is defined over).

Example: An ordinary-looking coin is tossed N times. $f_N(H)$ is the frequency of heads in N tosses, $f_N(T)$ is frequency of tails. The relative frequency estimates of $p(H)$ and $p(T)$ are:

$$p_N(H) = f_N(H)/N,\ p_N(T) = f_N(T)/N.$$

Suppose $N = 10$ and $f_{10}(H) = 7$, $f_{10}(T) = 3$. Let the ultrasmooth distribution be the uniform distribution, u:

$$u(H) = u(T) = 1/2.$$

Let λ equal 1/5. Then $\tilde{p} = 1/5\,p_{10} + 4/5u$:

$$\tilde{p}(H) = 27/50,\ \tilde{p}(T) = 23/50.$$

λ is usually selected in such a way that, all else being equal (e.g., number of elementary events),

$$\lambda \to 1 \text{ as } N \to \infty.$$

For example, Fienberg and Holland [2] calculate λ so as to minimize the expected squared error, under a number of assumptions, when p_N is replaced by $\tilde{p}$.

An alternative to convex smoothing (but still a smoothing method in the sense of Titterington, to be shown below), is to marginalize and reconstruct under the assumption of (conditional) independence. For the coin toss example, where there is a single random variable, this is not possible. Consider instead the following experiment: A coin and tack are flipped

simultaneously. The tack can land either point up (U) or point down (D). Suppose that the results of 50 flips are:

Coin	*Tack*	$f_{50}(.)$	$p_{50}(.)$
H	U	5	0.1
H	D	10	0.2
T	U	15	0.3
T	D	20	0.4

Marginals:

Coin	$p_{50}(.)$	*Tack*	$p_{50}(.)$
H	0.3	U	0.4
T	0.7	D	0.6

Estimate $\hat{p}$ computed from marginals under (pretty safe) assumption of independence between coin and tack flips:

Coin	*Tack*	$\hat{p}(.)$
H	U	0.12
H	D	0.18
T	U	0.28
T	D	0.42

Notice that there is no $\lambda \in [0, 1]$ such that

$$\hat{p} = \lambda p_{50} + (1 - \lambda)u,$$

where u is the uniform distribution. For example,

$$\lambda 0.2 + (1 - \lambda)0.25 = 0.18$$

implies $\lambda = 7/5$. (Similarly for the 14 other ultrasmooth distributions over this set of events.) Notice also that, as measured by cross entropy (directed divergence, relative entropy), $\hat{p}$ is closer to u than is p_{50}:

$$d(\hat{p}, u) = \sum_t \hat{p}(t) \times ln(\hat{p}(t)/u(t))$$

$$= 0.103$$

$$< 0.106$$

$$= d(p_{50}, u)$$

Before going beyond the uninteresting case of two binary variables, some definitions should be given.

Variables (or attributes) will be said to have *domains*. The domain of the variable *Coin*, in the example above, denoted *dom*(*Coin*), is the set $\{H, T\}$. A set of variables has a domain also, the cartesian product, under an arbitrary but fixed ordering, of the domains of its elements. For example,

$$dom(\{Coin, Tack\}) = \{(H, U), (H, D), (T, U), (T, D)\}.$$

The probability distributions we will be considering will be defined over domains of sets of finite variables like $\{Coin, Tack\}$.

Example:

Type	*Plant*	*Defective*	$p_{100}(.)$
Chain	Lubbock	No	15/100
Chain	Lubbock	Yes	1/100
Chain	Waco	No	22/100
Chain	Waco	Yes	20/100
Sprocket	Lubbock	No	12/100
Sprocket	Lubbock	Yes	1/100
Sprocket	Waco	No	26/100
Sprocket	Waco	Yes	3/100

The set {*Type*, *Plant*, *Defective*} is the *scheme* for p_{100}. The scheme for a distribution p will be denoted $s(p)$. The set of all distributions over a scheme V (or, more properly, over $dom(V)$) will be denoted P_V; the uniform distribution over V will be denoted u_V.

Marginal distributions are formed by *projecting* p onto subsets of $s(p)$. For tuples $w \in dom(W)$ and $b \in dom(B)$, $B \subseteq W$, $w[B] = b$ iff w and b agree on all attributes in scheme B. The projection of p with scheme V onto $A \subseteq V$ is the distribution $\pi_A(p)$, where

$$\pi_A(p)(a) = \sum_{t \in dom(V),\, t[A]=a} p(t).$$

A *model* of a scheme V is a set $X = \{V_1, \ldots, V_m\}$ such that $\bigcup_{j=1}^{m} V_j \subseteq V$ and $V_i \not\subset V_j$ for all $i, j \in \{1, \ldots, m\}$. ($X$ will sometimes be referred to as a model of a distribution with scheme V.) If X is also a cover of V, then X is a *reduced hypergraph* over V. Normally, attention is restricted to reduced hypergraph models of a given scheme V, although we will want to utilize the set of all models.

A distribution with scheme V may be projected onto a model $X = \{V_1, \ldots, V_m\}$ of V to form a set of marginal distributions (probabilistic database)

$$\pi_X(p) = \{\pi_{V_1}(p), \ldots, \pi_{V_m}(p)\}.$$

Example:

Type	*Plant*	$\pi_{\{Type,Plant\}}(p_{100})(.)$	*Plant*	*Defective*	$\pi_{\{Plant,Defective\}}(p_{100})(.)$
Chain	Lubbock	16/100	Lubbock	No	27/100
Chain	Waco	42/100	Lubbock	Yes	2/100
Sprocket	Lubbock	13/100	Waco	No	48/100
Sprocket	Waco	29/100	Waco	Yes	23/100

Note that, for example, we could refer to

$$\pi_{\{Type,Plant\}}(p_{100})(\text{Chain, Lubbock})$$

as

$$p_{100}(\text{Chain, Lubbock}),$$

but there are advantages to viewing the marginal distribution as a distinct object.

An important partial ordering on models is the *refinement relation*. A model X is a *refinement* of model Y (and Y is an *aggregate* or *coarsening* of X), denoted $X \leq Y$, iff for each $V_x \in X$ there exists a $V_y \in Y$ such that $V_x \subseteq V_y$. For example, $\{\{A\},\{B,C\}\}$ is a refinement of $\{\{A,B\},\{B,C\},\{D\}\}$. The set of all models over V together with the refinement ordering is a lattice. Any pair of models has a greatest lower bound equal to their least refined common

refinement and a least upper bound equal to the most refined structure of which they are both refinements. The universal upper bound of the lattice of models over V is $\{V\}$; the universal lower bound is $\{\varnothing\}$. For the (sub)lattice of reduced hypergraphs, the universal lower bound is $\{\{v\}|v\in V\}$. Least upper and greatest lower bounds are straightforwardly computed:

$$lub(X, Y) = f(X\cup Y)$$

and

$$glb(X, Y) = f(g(X, Y)),$$

where $f(Z)$ eliminates all $W\in Z$ such that there exist $U\in Z$ for which $W\subset U$, and

$$g(X, Y) = \{V_x\cap V_y|(V_x, V_y) \in X\times Y\}.$$

Examples:

$$lub(\{\{Type, Plant\}\}, \{\{Type\},\{Plant\},\{Defective\}\}) = f(\{\{Type, Plant\},\{Type\},\{Plant\},\{Defective\}\})$$

$$= \{\{Type, Plant\},\{Defective\}\}.$$

$$glb(\{\{Type, Plant\}\}, \{\{Type\},\{Plant\},\{Defective\}\}) = f(g(\{\{Type, Plant\}\}, \{\{Type\},\{Plant\},\{Defective,$$

$$= f(\{\{Type\},\{Plant\},\varnothing)\})$$

$$= \{\{Type\}, \{Plant\}\}.$$

The *extension* of p to the set $V\supseteq s(p)$ is the set of all preimages of p under the mapping $\pi_{s(p)}: P_V \rightarrow P_{s(p)}$.

$$E^V(p) = \{p'\in P_V|\pi_{s(p)}(p') = p\}.$$

The extension of a database D is the intersection of the extensions of its elements:

$$E^V(D) = \cap_{p\in D}E^V(p).$$

Thus, $E^V(\pi_X(p))$ is the set of all preimages of the database $\pi_X(p)$ under the mapping π_X; any model X of V partitions P_V into classes $E^V(\pi_X(p))$ equivalent with respect to projections onto X. Any $E^V(p)$ or $E^V(\pi_X(p))$ is the set of solutions to a system of linear equations and is therefore convex. An important special case is the extension of the projection of a distribution onto the model containing the empty set:

$$E^V(\pi_{\{\varnothing\}}(p)) = P_V.$$

Example: The projection of p_{50} onto the model $\{\{Coin\},\{Tack\}\}$ is

Coin	$\pi_{\{Coin\}}(p_{50})(.)$	*Tack*	$\pi_{\{Tack\}}(p_{50})(.)$
H	0.3	*U*	0.4
T	0.7	*D*	0.6

The extension of these marginals to the set $\{Coin, Tack\}$,

$$E^{\{Coin,Tack\}}(\pi_{\{\{Coin\},\{Tack\}\}}(p_{50})),$$

is the set of solutions p to:

$$p(H, U) + p(H, D) = 0.3$$

$$p(T, U) + p(T, D) = 0.7$$

maximum entropy reconstruction from a model.) It is not symmetric. It violates the triangle inequality. However, it has important information-theoretic properties not possessed by metric distances [5]. Four useful properties it has that are specific to this context and that metric distances do not possess are the following:

Corollary 4: $X \leq Y$ implies $d(p, J^V(\pi_Y(p))) \leq d(p, J^V(\pi_Y(p)))$. □

Theorem 5: For all p and X,

$$d(p, J^{s(p)}(\pi_X(p)) = H(J^{s(p)}(\pi_X(p)) - H(p). \square$$

Corollary 6: $d(p, u_{s(p)}) = H(u_{s(p)}) - H(p)$.

Proof: $u_{s(p)} = J^{s(p)}(\pi_{\{\varnothing\}}(p))$. □

The projection-reconstruction technique is a member of a class of smoothing techniques that Titterington [1] refers to as *minimum penalized roughness* methods. A minimum penalized roughness estimate is a distribution

$$\hat{p} = \arg\min_{p \in s(p_N)} \Delta_1(p, u_{s(p_N)}) + k\Delta_2(p, p_N),$$

where $k > 0$ and the Δ_i are nondegenerate, but are not necessarily metric distances. For a suitable model X, let

$$\Delta_2 = \begin{cases} 0, & \text{if } p \in E^{s(p_N)}(\pi_X(p_N)) \\ \infty, & \text{otherwise.} \end{cases}$$

With Δ_1 equal to directed divergence, Δ_2 as defined above, and $k = 1$, it follows from Corollary 6 that

$$J^{s(p_N)}(\pi_X(p_N)) = \arg\min_{p \in s(p_N)} \Delta_1(p, u_{s(p_N)}) + k\Delta_2(p, p_N).$$

Corollary 7: $X \leq Y$ implies

$$d(J^{s(p_N)}(\pi_X(p_N), u_{s(p_N)}) \leq d(J^{s(p_N)}(\pi_Y(p_N), u_{s(p_N)}). \square$$

From Corollary 7, the more refined a model is, the smoother (as measured by directed divergence between it and the uniform distribution) is the result of projecting the observed relative frequency distribution onto it and joining. Corollary 7 has the same form as the following statement of the smoothing effect of convex smoothing:

Theorem 8: $\alpha \leq \beta$ implies

$$e(\alpha p_N + (1 - \alpha)u_{s(p_N)}) \leq e(\beta p_N + (1 - \beta)u_{s(p_N)}),$$

where e is euclidean distance and $\alpha, \beta \in [0, 1]$. □

Note that when $\lambda = 0$,

$$\lambda p_N + (1 - \lambda)u_{s(p_N)}) = u_{s(p_N)},$$

and when $\lambda = 1$,

$$\lambda p_N + (1 - \lambda)u_{s(p_N)}) = p_N.$$

Similarly, when $X = \{\varnothing\}$,

$$J^{s(p_N)}(\pi_X(p_N)) = u_{s(p_N)},$$

and when $X = \{s(p_N)\}$,

$$J^{s(p_N)}(\pi_X(p_N)) = p_N.$$

Techniques exist for selecting λ in such a way that, among other things,

$$\lambda \to 0 \text{ as } N \to 0.$$

For example, Fienberg and Holland [2] compute λ as

element. As is shown in [3,4], there is a strong analogy between this function and the *natural join* from relational algebra. Therefore, we will denote it J:

$$J(P) = \arg\max_{p \in P} H(p).$$

The operator symbol J will be overloaded and applied to databases, in which context it will be parameterized also. The expression $J^V(D)$, where D is a database, should be interpreted as

$$J^V(D) = J(E^V(D)).$$

When $V = \cup_{p \in D} s(p)$, the expression $J^V(D)$ can be simplified to $J(D)$.

Suppose that p is unknown but that $\pi_X(p)$ (or an estimate of $\pi_X(p)$) is available. Estimating p as $J(\pi_X(p))$ amounts to making assumptions of conditional independence similar to those encoded by Markov and Bayesian networks [4]. To illustrate, suppose that $D = \{p_1, p_2\}$. Then

$$J(D)(t) = p_1(t[s(p_1)]) \times p_2(t[s(p_1)]) / \sum_{c \mid c[s(p_1) \cap s(p_2)] = b[s(p_1) \cap s(p_2)]} p_2(c).$$

Example:

Type	*Plant*	*Defective*	$J(\pi_{\{\{Type,Plant\},\{Plant,Defective\}\}}(p_{100}))(t)$	
Chain	Lubbock	No	0.149	=(16/100×27/100)/(27/100+2/100)
Chain	Lubbock	Yes	0.011	
Chain	Waco	No	0.284	
Chain	Waco	Yes	0.136	
Sprocket	Lubbock	No	0.121	
Sprocket	Lubbock	Yes	0.009	
Sprocket	Waco	No	0.196	
Sprocket	Waco	Yes	0.094	

In what follows, all references to reconstructability will be with respect to the function J.

Corollary 3: If p is reconstructable from X and $X \leq Y$, then p is reconstructable from Y. □

Recall that our motivation is to study the performance of techniques for improving an estimate of a unknown probability distribution based on limited observations by replacing an observed relative frequency distribution with the join of its projection onto a suitable model. We also want to establish guidelines for selecting the model. The key to this process is *approximate reconstructability*. We will use directed divergence to measure the degree to which a distribution is reconstructable from its projection onto a model and to compare models with respect to the degree to which they capture the (approximate) dependencies among the variables. Recall that the directed divergence (cross-entropy, relative entropy) between distributions p_1 and p_2 is

$$d(p_1, p_2) = \sum_t p_1(t) \times \ln(p_1(t)/p_2(t)),$$

where $0 \ln 0 = 1$. For all p_1 and p_2, $d(p_1, p_2) = 0$ iff $p_1 = p_2$. So, p is reconstructable from X iff

$$d(p, J(\pi_X(p))) = 0.$$

In general (for equally refined models – see below), X will be considered a better model of p than Y is if

$$d(p, J(\pi_X(p))) < d(p, J(\pi_Y(p)))$$

Although directed divergence is nondegenerate, it is not a metric distance. It is not defined for all pairs of distributions in a given P_V. (But it is defined for any distribution and its

$$p(H, U) + p(T, U) = 0.4$$

$$p(H, D) + p(T, D) = 0.6$$

$$p(t) \geq 0.$$

(The equations imply that $\sum_t p(t) = 1$.) The distributions $\hat{p}$ and p_{50} are solutions, as are infinitely many other probability distributions (represented with infinite precision). Notice that p_{50} is the only element of

$$\{p | p = \lambda p_{50} + (1 - \lambda)u_V, \lambda \in [0, 1]\}$$

that is also an element of

$$E^{\{Coin,Tack\}}(\pi_{\{\{Coin\},\{Tack\}\}}(p_{50})).$$

For any $p \in E^{\{Coin,Tack\}}(\pi_{\{\{Coin\},\{Tack\}\}}(p_{50}))$,

$$\pi_{\{Coin\}}(p)(H) = 0.3.$$

For any $p \in \{p | p = \lambda p_{50} + (1 - \lambda)u_V, \lambda \in [0, 1]\}$,

$$p(H) = 0.3\lambda + 0.5(1 - \lambda),$$

for some $\lambda \in [0, 1]$. These conditions imply that

$$0.3 = 0.3\lambda + 0.5(1 - \lambda)$$

$$= -0.2\lambda + 0.5,$$

which implies $\lambda = 1$.

Theorem 1: $X \leq Y$ implies $E^V(\pi_Y(p)) \subseteq E^V(\pi_X(p))$.
Proof: The equations determining $E^V(\pi_X(p))$ are linear combinations of the equations determining $E^V(\pi_Y(p))$. □

A distribution p is *identifiable* from (its projections onto) a model X iff

$$\{p\} = E^{s(p)}(\pi_X(p)).$$

Corollary 2: If p is identifiable from X and $X \leq Y$, then p is identifiable from Y. □

Pairs (p, X) for which p is identifiable from X are rare. A weaker condition is reconstructability. For a function

$$f: 2^{P_{s(p)}} \rightarrow P_{s(p)},$$

with the property

$$f(P) \in P,$$

p is *reconstructable* from X, with respect to f, iff

$$f(E^{s(p)}(\pi_X(p))) = p.$$

(Clearly, if p is identifiable from X, it is reconstructable from X, with respect to any function f, but not conversely.)

There are a number of reasonable reconstruction functions, for example, the function that selects the centroid. However, we will utilize the function that selects the *maximum entropy* element, where the entropy of a distribution p is

$$H(p) = -\sum_t p(t) \times \ln p(t).$$

Since the extension of a set of distributions is convex, there is a unique maximum entropy

$$N/(N+k),$$

where

$$k = (N^2 - \sum_{t \in dom(s(p_N))} (Np_N(t))^2) / \sum_{t \in dom(s(p_N))} ((Np_N(t))^2 - (N^2 / |dom(s(p_N))|)).$$

This estimate minimizes expected squared error under various assumptions about the unknown distribution. It is called a "bootstrap Bayes" estimate since values of the observed relative frequency distribution are used to determine the parameter λ.

We want to do something analogous using the lattice of models of $s(p_n)$ vs. the interval [0,1] of values for λ. Specifically, we will develop an estimation technique for which p_N is replaced by $J(\pi_X(p_N))$ with the property that

$$X \rightarrow \{\varnothing\} \text{ as } N \rightarrow 0.$$

The degree of refinement of X will be based on (we expect) the relation between N and the size of $dom(s(p_N))$. From a set M of models at the same "level" of refinement (see below), X will be selected as

$$\arg\min_{Z \in M} d(p_N, J^{s(p_N)}(\pi_Z(p_N))),$$

where d is directed divergence. From Theorem 5, this can be simplified to

$$X = \arg\min_{Z \in M} H(J^{s(p_N)}(\pi_Z(p_N))).$$

Denote as p_∞ the "true" but unknown probability distribution generating the observed relative frequency distribution p_N. Why should we expect $J^{s(p_N)}(\pi_X(p_N))$ to be a good estimate of p_∞, for suitable X? Why should we expect it to be a better estimate (on the average, on some metric) than p_N? First, for any $X < \{s(p_\infty)\}$,

$$N/|dom(V)| > N/|dom(s(p_\infty))|,$$

for all $V \in X$. Second, it is not necessary that p_N be especially close to p_∞ in order for

$$\arg\min_{Z \in M} d(p_N, J^{s(p_N)}(\pi_Z(p_N)))$$

to equal

$$\arg\min_{Z \in M} d(p_\infty, J^{s(p_N)}(\pi_Z(p_\infty))).$$

The key difficulty will be identifying the set M within which to search for the best model of p_N. Define the *level*, $l(X)$, of a model X recursively as

$l(\{\varnothing\}) = 0.$

$l(X) = l(Y) + 1,$

where Y is any *immediate refinement* of X. Y is an immediate refinement of X if it is a refinement of X and there is no Z distinct from X and Y such that

$$Y \leq Z \leq X .$$

We will use the phrase "level n" in connection with a lattice of models to refer to the set of all X such that

$$l(X) = n.$$

For example, level 2 of the lattice of models of $\{A, B, C\}$ has three elements:

$$\{\{A\},\{B\}\}, \{\{A\},\{C\}\}, \{\{B\},\{C\}\}.$$

Level 6 contains only the model

$$\{\{A, B\},\{A, C\},\{B, C\}\}.$$

The experiments conducted had the following structure: For a given set of variables V, a large number of probability distributions p_∞, where $V = s(p_\infty)$ were randomly generated. For

each of various values of N (e.g., 15, 30, 60, etc.), each p_∞ was used to generate a single p_N. The procedure was as follows:

Repeat N times
  Generate $x \in [0, 1]$
    if $x \in [0, p(t_1))$ increment $f(t_1)$
    else if $x \in [p(t_1), p(t_1) + p(t_2))$ increment $f(t_2)$
      .
      .
      .
    else increment $f(t_{|dom(V)|})$
For all t, $p_N(t) = f_N(t)/N$.
For each level k of the lattice, search for X such that

$$X = \arg\min_{Z||(Z)=k} d(p_N, J^V(\pi_Z(p_N))).$$

  For this X, calculate $e(p_\infty, J^V(\pi_X(p_N)))^2$
For this pair (p_∞, p_N), record the level at which $e(p_\infty, J^V(\pi_X(p_N)))^2$ was minimum.

[NOTE: e^2 was used for two reasons:

1. d is not defined for all pairs of distributions.

2. The projection-reconstruction method was compared with Fienberg and Holland's method.]

What was explored was the relation between N, $dom(V)$ and the level at which the X minimizing

$$e(p_\infty, J^V(\pi_X(p_N)))^2$$

can be expected to be found.

The hope was that relatively sharp distributions over the set 0, 1, 2, $\cdots$, $l_{\max}$ would be discovered (percentage of cases in which best model was at level l, for a given V and N). Unfortunately, except for experiments in which p_∞ was generated in such a way that it was reconstructable from some model, the tendency was for either $\{\varnothing\}$ or $\{V\}$ (which gave as an adjusted estimate either the uniform distribution or the original relative frequency distribution, respectively) to dominate the models at intermediate levels of refinement. Although there are good reasons to expect actual distributions (vs. randomly generated distributions) to be reconstructable, or approximately so (see discussions by Simon, Jaynes and Pearl [6,7,8]), the results are somewhat disappointing.

Figure 1 shows results of an experiment involving a lattice of models that is a chain, for three binary variables. (Call them A, B, and C.) The models considered, in increasing order of refinement, were

$$\{\varnothing\}, \{\{A\},\{B\},\{C\}\}, \{\{A,B\},\{A,C\},\{B,C\}\}$$

In addition, the Fienberg-Holland estimate, p_{fh}, was compared with the reconstructions from these models. The Fienberg-Holland estimate can be viewed as a surrogate for the given relative frequency estimate, p_N (which equals the join of its projection onto $\{A,B,C\}$), since

$$p_{fh} \rightarrow p_N$$

as

$$N \rightarrow \infty.$$

In Figure 1, N is plotted against the frequency with which the reconstruction from each model is closer to p_∞ than are the reconstructions from the other models. For each value of N, 1000 distributions p_∞ were generated at random and a p_N generated from it.

Notice that for each value of N, the distribution that is most frequently closest to p_∞ is either the uniform distribution (join of projection of p_N onto $\{\varnothing\}$) or p_{fh}.

An interesting phenomenon that arose in all of the experiments in which one of the uniform distribution, p_N or p_{fh} always outperformed the other distributions, is that the value of N at which the transition from the uniform distribution to p_N or p_{fh} occurs is the value of N for which the performance of all the models is most similar. For the cases studied, this value of N is approximately

$$5 \times 2^{|dom(V)|/3 - 1}.$$

The data in Figure 2 represent experiments in which p_∞ was reconstructable from one of the 8 covers of $\{A, B, C\}$. Here, there are ranges of values for N for which each estimate outperforms the others.

II. Phase Transitions in Database Consistency Problems

Much work has been done in recent years on the problem of identifying hard instances of constraint satisfaction problems [9]. Most of these problems are NP-complete [10]. Although such problems (probably) require exponential time in the worst case, in practice, many instances can be solved reasonably quickly.

The research on locating hard instances of these problems has focussed on identifying critical parameters in the description of an instance. For example, for randomly generated instances of 3-SAT, the difficult cases are concentrated around a particular ratio of clauses to variables [11]. For the problem of determining whether a partially specified latin square can be completed, there is a value for the percentage of entries that are fixed in advance around which the difficult cases are clustered [12]. There tends to be an abrupt change in the performance of algorithms as one goes from instances characterized by values less than the critical value to instances with greater values. This behavior is analogous to a phase transition in a physical system.

We conducted experiments to locate difficult instances of the problem of determining whether partially specified database frequency tables are consistent. The experiments involved binary attributes and database schemes of a very restricted form, as described below. The general framework, for the case of 3 attributes, is as follows:

A_1	A_2	f_1	A_2	A_3	f_2
0	0	x_8	0	0	x_{12}
0	1	x_9	0	1	x_{13}
1	0	x_{10}	1	0	x_{14}
1	1	x_{11}	1	1	x_{15}

A_1	A_2	A_3	f_0
0	0	0	x_0
0	0	1	x_1
0	1	0	x_2
0	1	1	x_3
1	0	0	x_4
1	0	1	x_5
1	1	0	x_6
1	1	1	x_7

The f_i are frequency distributions. The constraints are:

$$x_i \geq 0$$

$$x_0 + x_1 + \cdots + x_6 + x_7 = N$$

$$x_8 = x_0 + x_1$$

$$x_9 = x_2 + x_3$$

$$\cdots$$

$$x_{14} = x_2 + x_6$$

$$x_{15} = x_3 + x_7$$

Note that these equations imply the constraints: $\sum_{i=8}^{11} x_i = N$, $\sum_{i=12}^{15} x_i = N$.

Zero or more of the marginal probabilities $x_8, \ldots, x_{15}$ have values preassigned to them. (The case in which all of the marginals have values assigned to them was proved in 1977 to be NP-complete [13].) None of the joint probabilities have values preassigned. The problem is to determine whether values satisfying the constraints can be found for the remaining variables.

The number of variables can be reduced. The marginal tables can be reexpressed as:

A_1	A_2	f_1	A_2	A_3	f_2
0	0	$x_0 + x_1$	0	0	$x_0 + x_4$
0	1	$x_2 + x_3$	0	1	$x_1 + x_5$
1	0	$x_4 + x_5$	1	0	$x_2 + x_6$
1	1	$x_6 + x_7$	1	1	$x_3 + x_7$

Notice that in each subdistribution, each of the x_i appears exactly once. (This is not the case for more general database schemes.) Also, for database schemes like this one, of the form

$$\{\{A_1, A_2, \cdots, A_{n-1}\}, \{A_2, A_3, \cdots, A_n\}\},$$

each sum

$$x_i + x_j$$

is of one of two simple forms:

a) $i \in \{0, 1, 2, \cdots, 2^{n-1}-1\}$ and $j = i + 2^{n-1}$

b) $i \in \{0, 2, 4, \cdots, 2^n-2\}$ and $j = i + 1$.

The maximum number of marginal constraints is always 2^n, which equals the number of joint tuples (i.e., the number of variables).

In the latin square completion problem, the parameters are the number of rows and columns and the number of preassigned entries. The number of different values for the entries is equal to the number of rows. For the 3-SAT problem, the parameters that can be controlled are the number of propositional variables and the number of clauses. For this problem, there are three parameters (vs. two): the number of attributes, the number of preassigned marginal frequencies, and the value of N.

It was decided at first to control for the interaction of N with the other two parameters by keeping the sum of the number of partitions of N into 2^n parts constant. For the numbers of attributes in the experiments conducted (n = 4,5,6,7), N = 32 kept this quantity approximately constant. (Note: A more "realistic" value might have been something like 5×2^n - the recommended minimum number of observations for the applicability of certain statistical procedures. However, the set of possible values {0,1,...,N} for the variables would have been too large to conduct the experiments. For SAT problems, the variables have only two possible values, vs. N+1. For the latin square problem, the number of values is equal to the number of rows, but results have been reported for squares with at most 20 rows.)

The difficult instances of latin square completion problems occur when approximately 43% of the elements are preassigned, regardless of the number of rows. Similarly, for 3-SAT problems, the hard cases arise when the ratio of clauses to variables is approximately 4.3, regardless of the number of variables. These problems lack an analogue of the global constraint that the sum of all of the variables equals N.

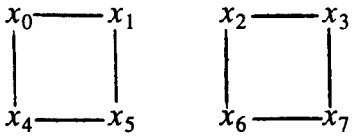
This combination of a relatively weak constraint involving all of the variables and constraints involving pairs of variables (values for specified marginal frequencies, equal to the sum of a pair of joint frequencies) complicated the search for a phase transition, i.e., a way of identifying, independently of the number of attributes, the point at which determining whether or not a pair of frequency tables is consistent becomes especially difficult.

It was decided to represent the constraint inherent (in an average sense - i.e., without considering the actual values of specified marginal frequencies) in an instance by means of an undirected weighted graph. Alternatively, the graph could be viewed as a multigraph, in which there is an edge between each pair of nodes (with the nodes representing joint frequencies, i.e., the variables) representing the weak constraint that they all sum to N, and edges connecting pairs of nodes that correspond to a marginal sum.

The subgraph representing the "strong", "local" constraints has an interesting structure.

There are 2^n edges and 2^n nodes. Thus, the number of these edges doubles as the number of attributes is increased by one, as does the number of edges in the complete graph. (The number of preplacements corresponding to the hardest cases less than doubles, however. Hence, something like the percentage of marginals specified will not characterize the phase transition.)

For 3 binary attributes, the "local" subgraph looks like this:



The edge from x_0 to x_1 represents the constraint

$$x_0 + x_1 = c.$$

If the marginal sum

$$x_0 + x_4 = d$$

is also given, this constrains the sum $x_1 + x_4$ also:

$$x_1 + x_4 = c + d - 2x_0.$$

Similarly for all the other "diagonal pairs" in the diagram above. Thus, when all marginals are specified, we should represent the constraint by the union of the complete graphs on these two sets of nodes. (More generally, we will want the transitive closure of the local graphs.)

So the situation of maximum connectivity will be represented by some combination of the complete graphs on disjoint sets of 4 nodes and the complete graph on all 2^n nodes. A weighted graph was formed by giving a weight of one for each edge from the complete "weak" graph and giving a weight of k to each edge from a local graph, where k was determined as discussed below.

For 5 attributes, the hardest instances occurred when 7 marginals were preassigned. For 6 attributes, 12 preassignments gave the hardest instances. With 5 attributes, the 7 marginal sums correspond to 7 edges to be distributed among 8 local graphs. Since this is fewer than one edge per graph, one cannot expect to need to add edges via transitivity. Similarly for 12 edges distributed among 16 local graphs, for 6 attributes. The following equation, in which the denominators characterize the maximum "weighted connectivity" and the numerators represent the weighted connectivity associated with 7 and 12 preassignments, for the 5 and 6 attribute cases, respectively, was solved for k:

$$\frac{16 \times 31 + 7k}{16 \times 31 + (8 \times 6)k} = \frac{32 \times 63 + 12k}{32 \times 63 + (16 \times 6)k}$$

They are equal (to 16.6016%) when $k = 427$.

Results for 7 and 4 attributes provide some support for this characterization of the constraint involved in these problems. For 4 attributes, with k = 427, 16.6% of "maximum weighted connectivity" is achieved when (to the nearest integer) 4 marginals are preassigned. This coincides with what was determined experimentally. Similarly, peak difficulty for the 7 attribute case is encountered when 16 marginal sums are specified, and this corresponds to 16.6% connectivity when k = 427.

Figure 3 plots median number of backtracks as a function of percentage of maximum weighted connectivity for 4, 5, and 6 attributes. The peaks occur at 16.6% connectivity for the 5 and 6 attribute cases. The peak occurs at 17.6% for 4 attributes. (It is impossible to achieve 16.6% connectivity for 4 attributes. The closest achievable values are 13.5%, corresponding to 3 marginal preassignments, and 17.6%, corresponding to 4 preassignments.) For 7 attributes, the percentage of runs terminated because the maximum allowed number of backtracks was reached is plotted as a function of connectivity. This peak was also achieved at 16.6% connectivity.

More experiments (involving higher cutoff thresholds, etc) will be conducted for the frequency version of the database consistency problem. Again, one interesting feature of the frequency version of the problem is the additional parameter, N, which complicated the description of the peak in solution difficulty.

In addition, a relational version of the database consistency problem will be investigated. The relational database consistency problem is easily represented as a satisfiability problem. To avoid 2-SAT, which is known to be solvable in polynomial time, the same type of database scheme will be used, but the attributes will be 3-ary instead of binary. The propositions corresponding to the consistency problem will be conjunctions of clauses containing three unnegated variables, but the clauses themselves will be either negated or unnegated. This will provide instances of SAT problems that correspond to an important practical problem, namely, determining the consistency of a database instance that may have become corrupted or that corresponds to a collection of incomplete partial views.

REFERENCES

[1] D. Titterington, "Common structure of smoothing techniques in statistics." *Int. Statistical Rev., v. 53*, 1985, pp. 141-170.

[2] S. Fienberg and P. Holland, "Simultaneous estimation of multinomial cell probabilities." *J. Am. Stat. Assoc., v. 68*, 1973, pp. 683-691.

[3] R. Cavallo and M. Pittarelli, "The theory of probabilistic databases." *Proc. 13th VLDB Conf.*, 1987, pp. 71-81.

[4] M. Pittarelli, "An algebra for probabilistic databases." *IEEE Trans. on KDE, v. 6*, 1994, pp. 293-303.

[5] J. Aczel and Z. Daroczy, *On Measures of Information and their Characterizations*. Academic Press, New York, 1975.

[6] H. Simon, "How complex are complex systems?" *PSA 76, v. 2*, 1976.

[7] E. T. Jaynes, "On the rationale of maximum-entropy methods." *Proc. IEEE, v. 70*, 1982, pp. 939-952.

[8] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Mateo, CA, 1988.

[9] T. Hogg *et al.*, "Phase transitions and the search problem." *Artificial Intelligence, v. 81*, 1996, pp. 1-15.

[10] M. Garey and D. Johnson, *Computers and Intractability*. W. H. Freeman, San Francisco, 1979.

[11] B. Selman *et al.*, "Generating hard satisfiability problems." *Artificial Intelligence, v. 81*, 1996, pp. 17-29.

[12] C. Gomes and B. Selman, "Problem structure in the presence of perturbations." *Proc. of the 13th Conf. on Uncertainty in Artificial Intelligence*, 1997.

[13] S. Reiss, "Statistical database confidentiality." Report No. 25, Dept. of Statistics, Univ. of Stockholm, 1977.

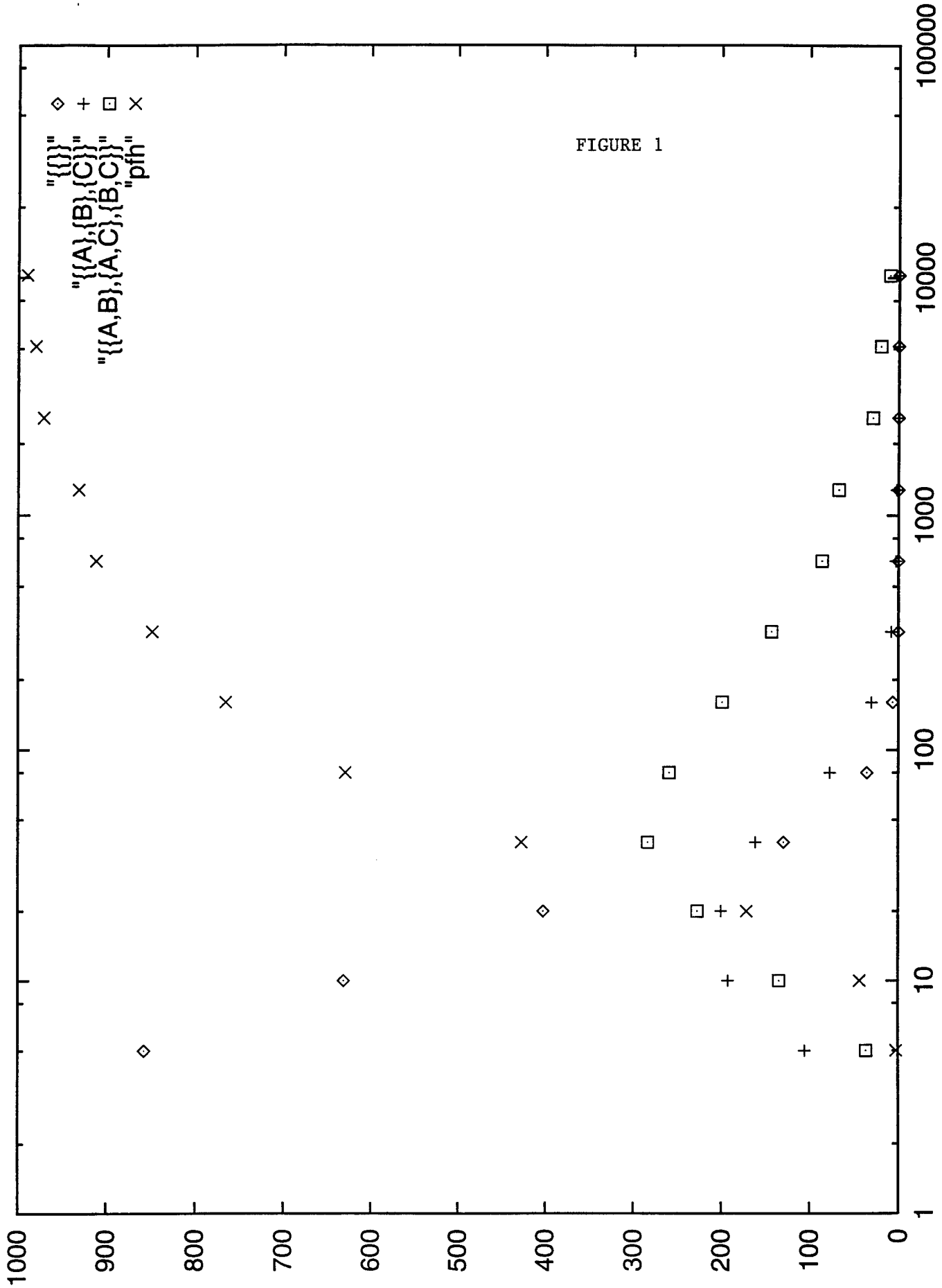

"{{}}"
"{{A},{B},{C}}"
"{{A,B},{A,C},{B,C}}"
"pfh"
1000
900
800
700
600
500
400
300
200
100
0
100000
10000
1000
100
10
1

FIGURE 1

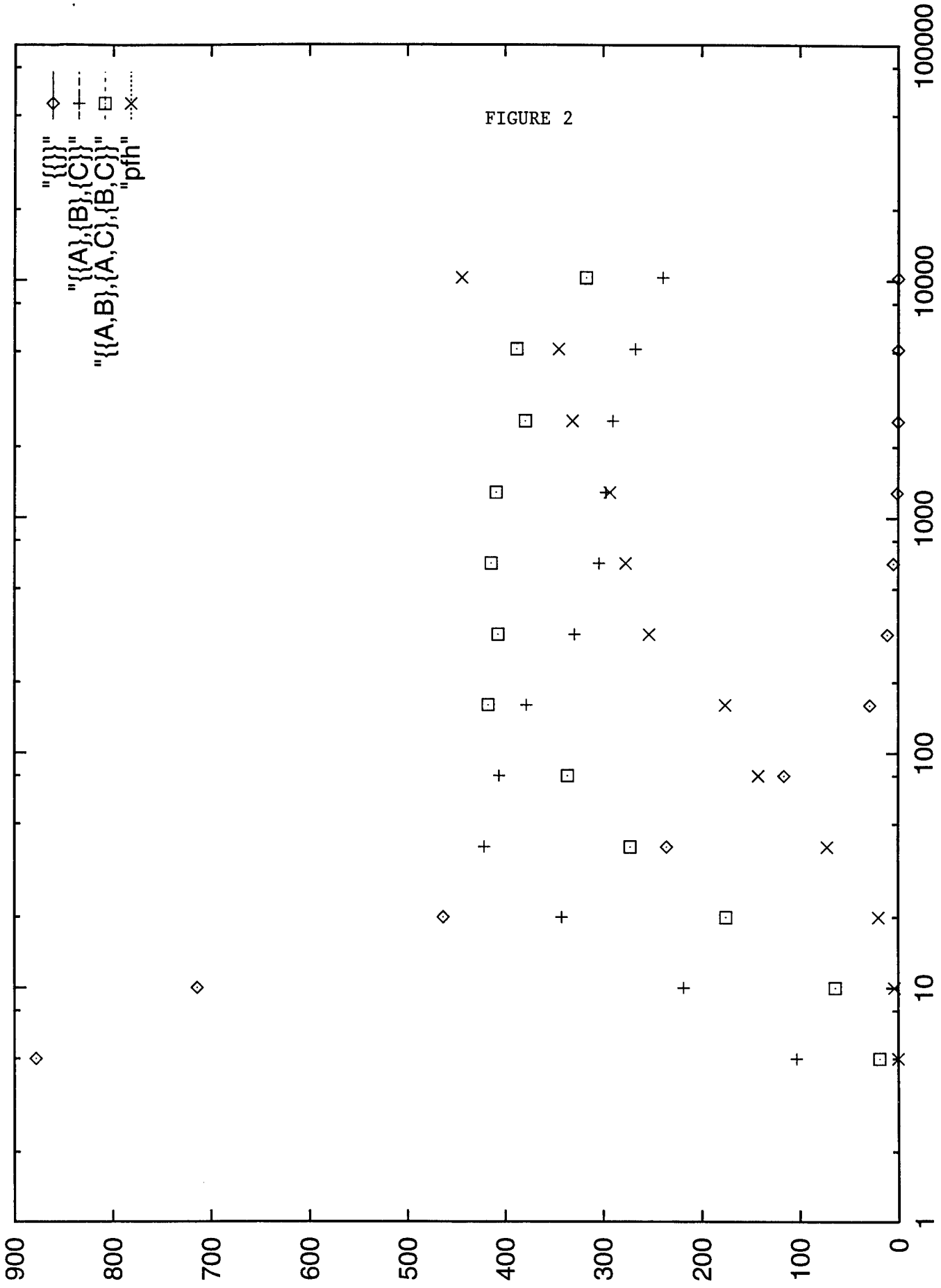

"{{}}"
"{{A},{B},{C}}"
"{{A,B},{A,C},{B,C}}"
"pth"
900
800
700
600
500
400
300
200
100
0
1
10
100
1000
10000
100000

FIGURE 2

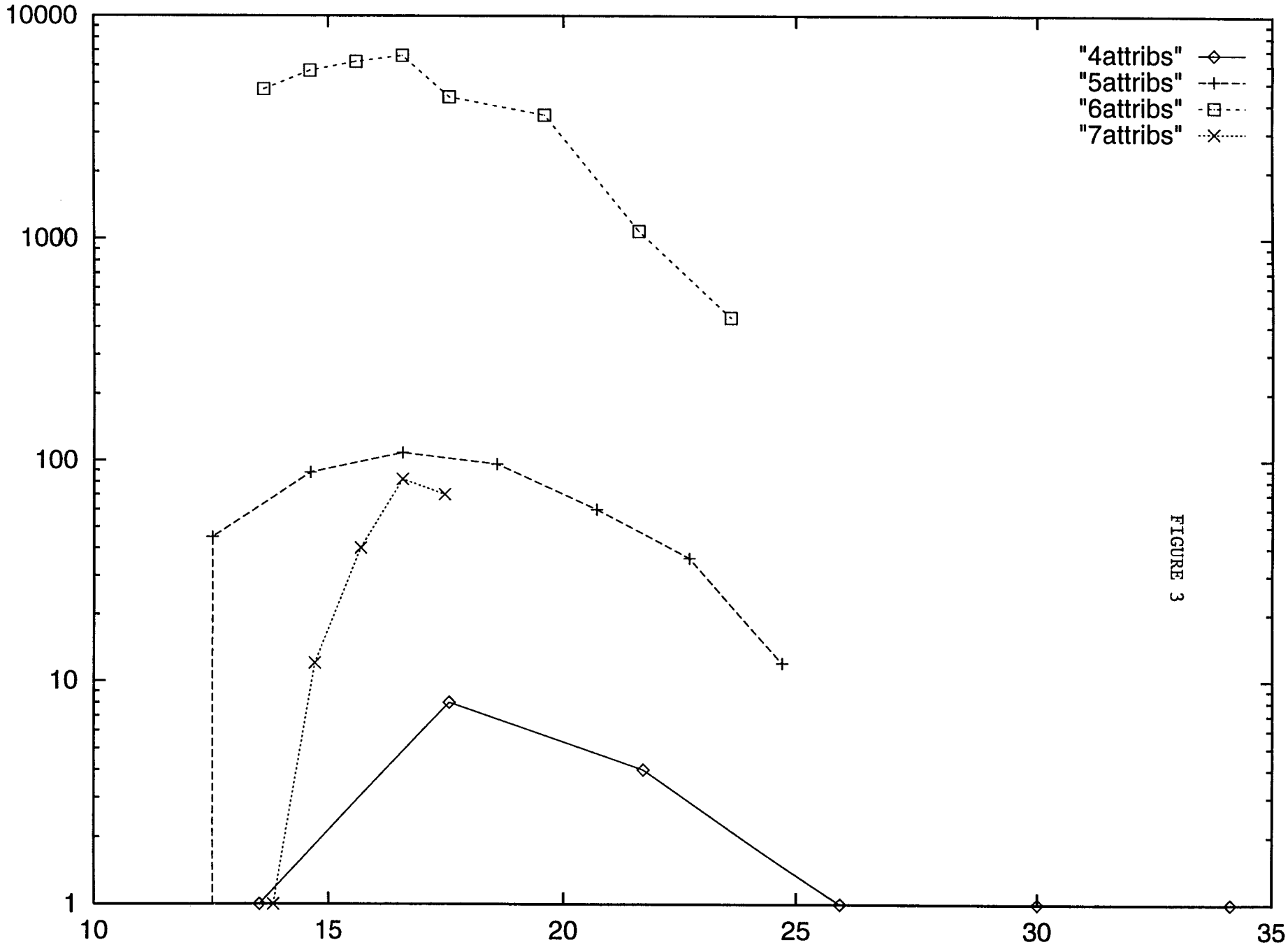

10000
1000
100
10
1
10
15
20
25
30
35
"4attribs"
"5attribs"
"6attribs"
"7attribs"

FIGURE 3

LOW DATA RATE MULTIMEDIA COMMUNICATION USING WIRELESS LINKS

Salahuddin Qazi
Associate Professor
School of Information Systems and Engineering Technology

State University of New York Institute of Technology
P.O. Box 3050, Utica, NY 13504-3050

Final Report for:
Summer Faculty Research Program
Rome Laboratory

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Rome Laboratory

September 1997

LOW DATA RATE MULTIMEDIA COMMUNICATION USING WIRELESS LINKS

Salahuddin Qazi
Associate Professor
Department of Electrical Engineering Technology
State University of New York Institute of Technology, Utica, New York.

Abstract

The report analyzes the variable length protocol medium resource controller(MRC) and a configurable protocol stack currently used for low data rate multi-media communications over wireless links at the Rome Air Force Lab, Rome, New York. The report also investigates the use of widely used TCP/IP protocols for such applications and discusses the methods of integrating the currently used proprietary protocols to the TCP/IP over wireless links.

LOW DATA RATE MULTIMEDIA COMMUNICATION USING WIRELESS LINKS

Salahuddin Qazi

Introduction

Wireless networks can be classified into networks consisting of mobile (cellular) wireless link, satellite links (geosynchronous and LEOS) and High Frequency links hopping through ionosphere. Apart from the difference in the range of frequencies and bandwidth used by these links, they also differ in their path lengths and propagation delay. Despite their differences they have many similarities so that the solutions used to improve the performances of wireless links may be applicable to all these links. These links present a lossy and highly unsymmetric end to end communication channel, giving rise to propagation delays, bit errors , limited bandwidth and intermittent connectivity.

There has been a rapid development [1] in the last decades for HF modem technology to adapt to the challenges of HF skywave channels which has been used for data communications in military applications. The effective use of these modems for reliable data delivery, however, has resulted in to a number of Federal and military standards and implementation of proprietary protocols . In order to increase the range of services and applications like in Tele-medicine, it is important to be connected or integrated to the widely used universal TCP/IP protocols. The purpose of this report is to analyze a set of proprietary protocols called Media Resource Controller(MRC) and configurable protocol stacks (CPS) currently used at the Rome Air Force Lab, New York. Use of TCP/IP for low data rate communications over wireless links and integration of these proprietary protocols with TCP/IP is also investigated.

Medium Resource Controller (MRC) over wireless links

Media Resource Controller is a communication facility which employs a variable packet length protocol for low data rate communications on half duplex/ full duplex RF links including HF, VHF, UHF, SAT COM, Land line, and serial media. It has been demonstrated to transmit E-mail, patient vital signs, full color hi-resolution scope images and EKG in air to ground and ground based communication systems. The media resource controller is a proprietary protocol based on open system architecture and requires each node to have a similar protocol for communications. It provides reliable connection-oriented service for low data rate links using modems and employs a character-oriented idle RQ (stop and wait) protocol. The protocol has been written in MS -DOS, UNIX and Microsoft Windows.

High frequency (HF) communication channel is considered to be one of the most difficult communication channels because of severe frequency selective fading due to ionospheric multipath propagation and the interference from other users. The MRC protocol incorporates MIL-STD-188-141A [2] conformed Automatic Link Establishment (ALE) controller to detect and avoid slowly changing interference and fading. Its primary objective is to provide a capability to maximize the probability of establishing a usable HF link between the communicating stations [3]. The ALE controller employs 8-ary FSK modem, half-rate Golay forward error correction (FEC), interleaving , and triple-redundant/majority-vote coding to communicate the basic protocol unit called an ALE word. Its features include auto call, manual call ,link quality analysis, frequency tuning, configuration programing and HF network administration. The land line telephone contain the features like auto dial, alternate auto dial and auto baud. It is capable of automatic switching to alternate RF media and a higher throughput without any loss or interruption of data flow. It is also capable of alternate routing by using adaptive alternate path algorithm based on link quality and hop count.

Data Interfacing and Formatting

The actual transfer of data takes place with the help of user interface program called the Advanced Multimedia Information Distribution System (amidas). It is implemented on the SUN SPARC work station, using UNIX system 5 operating system. The data is segmented into blocks of 128 bytes packets. Function of segmentation is similar to the basic function of transport layer where it accepts data from the session layer , splits it up into smaller units and passes it to the network layer. The size of data packets are of variable length and ranges between 128 bytes to 4096 bytes, depending on the type of media and quality of link. If the link is good, the data packet will have larger packets rising to the maximum packet size as shown below:

Media	Max Packet sizes	Minimum packet size	Increment
HF	1024	128	128
VHF	2048	128	512
UHF	4096	128	512
SER	4096	128	512
LL	2048	128	512

Each block of data starts with two bytes indicating the number of blocks and ends with two bytes of cyclic redundancy check (CRC). This insures that in case of repeated crashes for lengthy file transfer with a certain time between crashes, only the data transferred after the last checkpoint have to be repeated.

CRC checks the received data after every 128 bytes and only those packets are re-transmitted which are received in error based on the value of received CRC. Packets are assembled and disassembled automatically. Packet retries differ from different media as given below:

Media	Packet retries	Abort retries
HF	5	2
VHF	5	2
UHF	5	2
SER	3	2
LL	3	2

CRC is calculated on the receiving end and its evaluation is sent back to the sender as a 4 byte character like 1 , !F or FF as CRC evaluator. There is no error check on the sender side.

MRC Header

The segmented data from the transport layer of MRC protocol is interfaced to the network layer where a packet has to travel from one network to another to get to its destination. Here the key design issue is to determine how packets are routed from source to destination. The MRC has a limited number of nodes , hence this layer is fairly thin and is combined with data link layer. The task of data link layer is to transform a raw transmission facility into a line free of undetected transmission errors. This is accomplished at the transmitting end by breaking the input data up into data frames , transmitting them sequentially, and processing the acknowledgment frames sent back by the receiver. Its function also includes the grouping of bits of the physical layer into frames, dealing with transmission errors and regulating the flow of frames so that slow receivers are not overwhelmed by the fast transmitters.

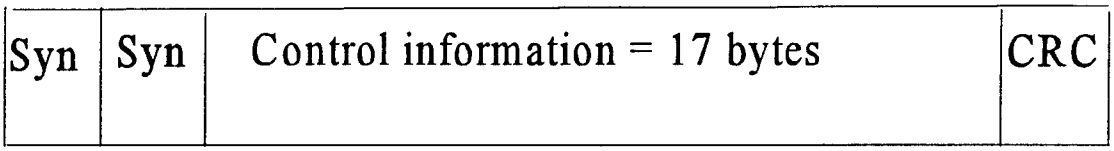
MRC header size is normally 20 bytes long and it can also be of variable size. The header packet consists of two bytes of SYNC located in the start and one byte of CRC at the end of the header. Synchronization takes place with two bytes of information (7e 81).There are other control bytes such as:

Start of packet header (one byte) and complement of start flag (one byte) which is used to synchronize the header. Packet type (one byte) indicates the type of packets, namely data packet, message packet, routing information packet. Source node (one byte) and destination node (one byte) identifies the source and destination. Route to node (one byte) and Route from node (one byte) gives network routing information. Priority (one byte). There are four levels of priority

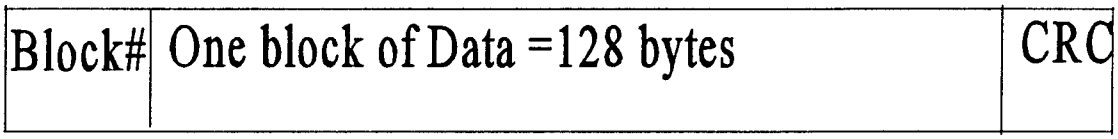
ranging from #1=Routine, 2=Priority, 3=Immediate, 4=Flash. Higher priority goes first by interrupting lower priority. The interrupted lower priority traffic will continue after the higher priority is completed at the point, it was interrupted. Block acknowledgment bit string (four bytes) is used only when data packet is transmitted. Acknowledgment packet is received if data is received. No acknowledgment packet is sent if no data is received and the sender is waiting. Wait Ack Timeout (the time transmitter has to wait for acknowledgment) is variable. In case of HF link it is 10 seconds. There is also receiver timeout which is 6 seconds for HF links

File sequence number, packet size, packet number, header CRC, all are two bytes each.

Flow control is not very critical due to the type of traffic present in the system. The buffer size is between 128 bytes to 4 K bytes.



MRC Header= 20 bytes, Syn=1 byte, CRC=1 byte



Data= 128-4096 bytes, Block No=1 byte, CRC=1 byte

MRC Frame

As the physical layer transmits only stream of bits without any knowledge of its structure, it is up to data link layer to create and recognize the frame boundaries. This is done by attaching special bit patterns to the beginning and end of the frame. It consist of three bytes 7e 81 (0-256) at the start and two bytes of 7e 81 at the end of the frame. If these bit patterns appears in the data , special care must be taken to make sure these patterns are not incorrectly interpreted as data frames. This is usually accomplished by way of character stuffing.

The handshaking is accomplished with the hardware, though SER communication uses software handshaking at the rate of 19,200 bauds. Each node listens to the other node and transmits only, when no other node is transmitting. However , if there is an urgency and while listening to other, the station with less priority will be told to hold back to allow station of higher priority to communicate. It is essentially a collision detection system and retransmission is needed upon collision. There is no routing as point to point communication is the predominant mode.

The Physical Layer

The physical layer is concerned with the mechanical, electrical and procedural interfaces between the user equipment and network terminating equipment. It also addresses the issues of physical medium, which lie below the physical layer and how the initial connection is established and terminated and whether transmission may proceed in both directions simultaneously.

Number of media as mentioned earlier have been used for MRC communication facility and it is accessed through a modem. Each modem operates at a certain baud rate and requires certain number of bits, a parity and stop bit as shown below:

Media	Baud rate	parity	no of bits	stop bits
HF	2400	N	8	1
VHF	9600	N	8	1
UHF	2400	N	8	1
SER	19200	N	8	1
LL	4800	N	8	1

TCP/IP Translation

An MRC can be translated to TCP/IP by using suitable software implemented on a PC, which will discard the header and trailer of MRC and add TCP/IP header and trailer. Further it will descend to the remaining layers of TCP/IP and makes its way to the TCP/IP network. This means that we have to design TCP/IP header based on the information extracted from the MRC header and trailer.

The other way to convert a protocol is to keep the header and trailer of MRC and add a new header of the new protocol (TCP/IP in this case). It is called encapsulation where each machine supporting the TCP has either a user processor or part of the kernel that manages TCP streams and interfaces to the IP layer. A TCP entity accepts user data streams from local processes, breaks them up into pieces of about 1500 bytes (in practice), and sends each piece as a separate datagram for transmission. On the receiving end, the received IP datagram containing TCP data is reconstructed by the TCP entity.

TCP/IP Integration By Way of Configurable Protocol Stack (CPS)

The difference between translating a protocol and CPS approach for TCP/IP is that CPS adds extra stacks of IP and TCP on the existing stacks without stripping the header and trailer. Since CPS also contains some functionality in the transport and network layers, the addition of TCP and IP gives some duplication by the stacking method

Configurable Protocol Stack (CPS)

CPS is a configurable protocol whereby additional stacks can be introduced to increase its functionality. In that sense it is a flexible protocol and can incorporate and integrate new protocols like ATM and TCP/IP with suitable layers. CPS layers are normally located below TCP/IP stacks, although it has some functionality at the session layer. It is a proprietary protocol and is based on open system architecture. In the currently used CPS protocols TCP and IP layers are configured with stacks. The applications are interfaced to the TCP stack by way of BSD socket API interface. It uses SUN SPARC work station's kernel to configure various protocols stacks and provide gateway for Internet working. One workstation is used for each node. It is a collision avoidance system where the collision are avoided by token management technique, so that two stations cannot communicate simultaneously. The acknowledgment is piggybacked in a full duplex system. The flow control is application dependent and the buffer size is 100 byte.

The composition of its header and data format is similar to the MRC as explained above. The difference lies in its accessing technique which is like a token ring and its flexibility to be integrated with other protocols by the addition of stacks for various functions.
CPS operation is divided into five functionality.

1. Synchronization
2. Segmentation
3. Network

4. Stop and Wait , CRC and Framing
5. Physical layer

Physical layer is a serial layer which is connected to null modem. The null modem is a simulated link for connecting the transmitter to the receiver and receiver to the transmitter. Radio link is a logical link and serial connection is the physical connection.

TCP/IP Header

TCP is a connection-oriented protocol designed to provide end-to-end byte stream over unreliable links. All TCP connections are full duplex and point to point. Copies of every transmitting messages are kept, making it a slow system. The basic transmission unit of TCP is called a segment which may or may not contain data. Each segment has a header that contains six 32-bit words or a fixed 20-byte header plus an optional part as shown in the figure. The segment size is limited by the IP pay-load of 65,535 bytes and the maximum transfer unit (MTU) of the network. Larger segments can be broken up into multiple segments by a router of a network that it must pass through. TCP uses sliding window protocol by starting a timer when it transmits a segment. The receiver sends back a segment with or without data on bearing an acknowledgment equal to the next sequence number it expects to receive, If the sender does not get any acknowledgment in the given timer (window) time, it will transmit the segment again. Further details are given in references [4] and [5].

TCP Over Wireless Links

TCP was developed originally for wired networks and provides error-free flow of packets by adaptive retransmission and congestive control mechanism. The retransmission mechanism is based on timers to detect when a packet has not been acknowledged by the receiver. Transmission time-out (RTO) is usually calculated conservatively to avoid unnecessary retransmission as all unacknowledged packets are re-transmitted after the expiration of the timer. The inherent problems of wireless links like propagation delay, fading and loss result in degrading the TCP performance over such links and rate of packet loss may therefore be considerably higher than wired networks.
On potentially congested networks, it is necessary for TCP to implement congestion avoidance algorithm. These algorithms assume that time-outs are caused by congestion and not by lost packets. In order to alleviate the congestion , the TCP slows down and sends less vigorously to reduce the network load Jacobson's slow start algorithm [6]. As the wireless link loses packets all the time , the proper approach of dealing with lost packets is to retransmit these packets again as fast as possible Slowing down as in

congestion control makes matters worst as it reduces the throughput. Number of researchers have attempted to improve the performance of TCP over wireless channels by proposing their solutions operating at the transport layer.

Bakne and Badrinath[7] proposed a solution for improving the performance of TCP by splitting the TCP connection (Indirect TCP) into, two separate connections, one between mobile host (sender) and the base station and the other between base station and the stationary host (receiver). The base station simply copies packets between the connections in both directions. It attempts to isolate the dynamics of TCP's congestion control from the interference of losses on the wireless channel. The time-outs in the first connection can slow the sender down, whereas timeout on the second one can speed up making both connections homogeneous. This however violates the semantics of TCP as each part of the connection is a full TCP connection.

Balakrishnan et al [8] achieved similar results without splitting a TCP connection but adding agents that observe and caches TCP segments going out to the mobile host, and acknowledgment coming back from it. It effectively works by keeping a log of unacknowledged TCP packets at the base station and re-transmitting them to the mobile host. One disadvantage of this method is that if the wireless link is very lossy, the source may time-out waiting for an acknowledgment and invoke the congestion control algorithm. In case of indirect TCP method, the congestion control algorithm will never be started unless there is a congestion in the wired part of the network.

Pravin Bhagwat et al [9] proposed link layer, instead of transport layer retransmission mechanism to improve the performance of TCP in wireless LAN. They argued that error behavior in wireless LAN is often bursty and most retransmission attempts to recover errors fail , causing poor channel utilization. They also observed that in case of multiple session sharing a wireless link FIFO packet scheduling can cause HOL blocking effect resulting in unfair bandwidth allocation among the various sessions. Based on these observation the authors proposed channel state dependent packet scheduling (CSDP) methods to overcome these problems.

TCP protocols for HF and satellite links

Satellite links and High Frequency links are known for longer transmission paths, higher propagation delay and errors. The satellite link has a constant propagation delay of up to 1 second depending on the number of hops, while HF half duplex networks consist of slightly larger variable propagation due to the link turnaround time. Packet loss due to bit errors in a satellite network is not as serious a problem as HF networks, which can observe significant bit errors due to the phenomenon of channel fading.

Vivek Arora et al [10] proposed the use of TCP/IP spoofing and selective acknowledgment dropping to increase the throughput of a hybrid satellite-terrestrial networks. Their approach of improving the throughput is based on keeping the window size of TCP for flow control the same and not modifying the existing protocols at the end hosts. Their hybrid Internet access system has two network interfaces, one attached to a receive -only VSAT via a special bus PC adapter and the other is a modem attached to a serial port. The hybrid terminal uses a modem connection for outgoing traffic while receiving incoming information through the VSAT. The problem of bandwidth-delay is solved by transparently splitting the end-to-end TCP connections into two parts consisting of conventional terrestrial portion and the hybrid portion. Thus isolating the satellite channel which has a long delay from Internet host having a small default window at the hybrid gateway improves the throughput. The problem of congestion in the low-bandwidth upstream path resulting due to many acknowledgments generated in the large -bandwidth downstream path is solved by selectively dropping redundant acknowledgment packets.

Robert C. Durst et al [11] jointly with NASA and the U.S. DoD have designed, specified, implemented a set of protocols for use in space data communications, known as Space Communications Protocol Standard (SCPS-TP). It consists of standard TCP with a set of extensions for implementation and enhancement changes which addresses the problems of wireless links. The authors identified that there are three sources of data loss in space links, namely congestion, corruption and link outages . Performance of the network were improved by developing mechanism to detect the sources of data loss and responding appropriately by using SCPS-TP Selective Negative Acknowledgment option. TCP's acknowledgment strategy was also modified to accommodate asymmetric channels by sending ACKs much less frequently and using TCP vegas congestion control mechanism to cope with the reduced ACK frequency.

Alan Ulrich Kennigton [12] of DSTO studied the performance of FTP/TCP by simulating delays and error links and analyzed it with delay, packet loss and loss added delay. He compared TCP/FTP with UDP based FITFEEL 0.0 protocol and confirmed that FTP/IP because of its ARQ and other inherent delays is not very suitable to use on wireless links. The author found that there is drastic bit reduction when propagation delay is involved during file transfer. He confirmed from his experimental results that bit rate falls as the reciprocal of the delay. In case of packet loss in HF links the effect on file transfer time is significantly increased. He transmitted packet voice application program at 4800 bits/sec by transmitting small UDP packets without any ARQ protocol. The transmission was so successful that in spite of large packet loss , it was often used for communicating with the ship instead of using SMTP. The author further used a selective ARQ scheme with fixed -size data packets of about 100

Bytes, ACK packets containing lists of correctly received file segments, and periodic transmission of file attributes. Such a protocol called FITFEEL 0.0, took about an hour instead of 3 days to transmit 50 K bytes.

Samaraweera et al [13] proposed the modification of TCP transport protocol in the end system to improve the performance of TCP for a satellite link using a terminal with a small aperture. They argued that estimating the correct value of RTO is an important factor in providing efficient TCP error recovery and is usually calculated from the measured round trip time (RTT). In most TCP implementations , it is measured by using a tick of 500 ms provided by the operating system. The tick increments a counter starting from the time a packet is sent to the tick after the corresponding acknowledgment is received. Increase in traffic causes queuing delays in routers and variance in the RTT.
The author suggests a modification in the transmitting host without adding any data to the transmitted packets. This is achieved by introducing an algorithm, recorded packet leaving time (RPLT) based on using the transmission time of each segment for which an acknowledgment is received. The transmitter records the packet leaving time in the first transmission using the system clock as well as noting the sequence number of the first octet in the segment. Upon acknowledgment, the transmitter searches the list of sequence number to select the latest segment with a sequence number which is less than the acknowledgment number. Transmission time associated with this segment is used to calculate the RTT. All the list entries with a sequence number older than the selected one are cleared. The algorithm continues to measure the RTT by monitoring acknowledgment to first time transmission, while the packets are being transmitted.

The second proposal by the authors is to use a non-congestion algorithm to improve the performance of TCP for wireless links. As the algorithm used in congestion avoidance control in potentially congested network normally uses the same timers for error recovery and congestion avoidance this results in to interaction between the two timers and causes TCP to perform poorly over wireless links. The author proposes a non-congestion packet loss indication to reduce the interaction, hence loss of errors. This error can be reduced by using fast retransmission and fast recovery algorithm to reduce the interaction between congestion control and error recovery. A receiver in an TCP protocol normally acknowledges received segments even when packet loss results in the same acknowledgment value being sent. Jacobson [6] suggested that a transmitter interprets three consecutive acknowledgments for the same segment as indication of a packet loss and retransmits the first unacknowledged segment immediately. This will reduce the congestion window without performing the start phase. Brakmo et al [14] argued that most packet losses are either much larger or the window is to small , that the transmitter will never receiver three duplicate

acknowledgments. They used time stamped technique to accurately measure the round trip time (RTT) and implemented the algorithm by re-transmitting the unacknowledged segment when a duplicate acknowledgment is received. The techniques based on duplicate acknowledgment works well provided there is only one packet loss per transmission window. In case of TCP applications like FTP, a large TCP window size is used and many packets could be lost within the transmission window giving rise to slow start congestion recovery. Samarweera et al [13] proposed a non-congestion packet loss indication algorithm (NCPLI), which uses a feedback mechanism to inform the receiver of packet losses which are not due to congestion. Implementation of such an algorithm requires one additional bit to be used in the header of all TCP packets.

Conclusion

It is evident that of all the wireless links reviewed here, the half duplex HF wireless link suffers the most with packet loss due to bit errors and large propagation delay. The flow control of TCP/IP protocol depends on the round -trip-time and window size. As HF link is a segment of wireless TCP/IP network, the round trip time (RTT) grows very large, so that the end hosts must wait a longer time for an acknowledgment from the other end. Large value of RTT can be compensated by increasing the TCP transmit window size but that will cause the transmitter to wait an unnecessarily long period before transmitting a lost packet. In case of long waiting for the acknowledgment the time-outs of the timer will invoke a congestion control, hence reducing the congestion window and in return reducing the throughput by the loss of a packet. It is not surprising that researchers like Alan Ulrich have suggested the use of UDP based (non TCP protocol) protocol for low data rate HF links.

A substantial part of the work discussed is in progress. Improvements to the performance of TCP/IP over wireless networks is of great interest in wireless mobile communications . Of the two most promising solutions namely Indirect - TCP (Bakne and Badrinath) and Snoop protocol (Balakrishnan), HF links can improve performance by using Indirect-TCP or its modification. This method can also accommodate end-to-end encryption mechanisms . The use of non-congestion algorithms and recorded packet leaving algorithms (Samaraweera) also seem promising and need to be further investigated. Reducing total number of acknowledgment or (using only selective acknowledgment as in the Configurable Protocol Stack (CPS) protocol) and packets of fixed bytes are also likely solutions which can improve the performance of TCP/IP over half duplex HF wireless networks.

The use of UDP protocols (like FITFEEL 0.0) can be used to send FTP much faster than sending encapsulated IP datagrams without establishing a

connection. As UDP does not use acknowledgment and does not provide feedback to control the rate at which information flows between the machines, the messages can be lost, duplicated, or arrive out of order.

Acknowledgments

The author would like to thank the Air Force office of Scientific Research, Bolling AFB Washington, DC, and Air Force Lab, Rome, for enabling the project under the summer Research Faculty program. Special thanks are due to Brian Spink, Joe Macini, Larry Spadaro, Sean and Rich Brady for numerous discussions and help.

References

1. Eric E. Johnson, "Asymptotic Throughput of the FED-STD Data Link Protocol", PP 700-705, 1996 IEEE.
2. MIL-STD-188-141 A, "Interoperability and Performance Standards for Medium and High Frequency Radio Equipment," U.S. Army Information Systems Engineering Command, Fort Huachuca, Arizona, 19923.
3. Richard Lay, "Error Correction in High Frequency Automatic Link Establishment Radios With And Without Link Protection" PP 696-699,IEEE 1996.
4. "Computer Networks" by Andrew S. Tanenbaum, Third Edition, 1996 Prentice Hall, Inc, New Jersey 07458.
5. "Internetworking with TCP/IP", by Douglas E. Comer, Volume 1, Principles, Protocols, and Architecture, Third Edition, 1995 Prentice Hall, N J 07458.
6. Jacobson, V., "Congestion Control and Control," Proceeding SIGCOMM '88 Conference ACM,PP. 314-329, 1988.
7. Bakne A., and Badrinath, B.R., "I-TCP., Indirect TCP for Mobile Hosts," Proceeding of Fifteenth International Conference on Distributed Computer Systems, IEEE, pp136-143, 1995.
8. Balakrishnan, H., Seshan,s,and Katz, R.H., "Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks,"Proceeding ACM Mobile Computing and Networking Conference., ACM, pp. 2-11, 1995.
9. Pravin Bhagwat et al, "Using Channel dependent packet scheduling to improve TCP throughput over wireless LANs", Wireless Networks , pp 91-102, 1997.
10. Vivek Arora et al, "Asymmetric Internet Access Over Satellite-Terrestrial Networks", American Institute of Aeronautics and Astronautic, pp.476-482, 1995
11. Robert C. Durst et al. "TCP Extension for Space Communications ", US Government Contracts DAAB0796-C-E601,F19628-94-C-0001,and NAS532607. 1996.

12. Alan Ulrich Kennigton, "The FITFEEL Transmission Protocol", Defence Science and Technology Organization, Salisbury, Australia, 1997.
13. Samaraweera, N., Fairhurst G., "Explicit Loss indication and accurate RTO estimation for TCP error recovery using satellite links",IEE proc. Commun Vol 144, No. 1, pp. 47-53, February 1997.
14. Brakmo S. Lawrence, Peterson L. Larry, "TCP Vegas: End to End Congestion Avoidance on a Global Internet" IEEE Journal on selected Area in communications, Vol 13, No.8, pp.1465-1480, October 1995.

AN IMPLEMENTATION OF THE MESSAGE PASSING INTERFACE ON RTEMS

Arindam Saha
Associate Professor
Department of Electrical & Computer Science

Mississippi State University
P.O. Box 9571
Mississippi State, MS 39762

Final Report for:
Summer Research Program
Rome Laboratory

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, Washington, DC

And

Rome Laboratory

August 1997

AN IMPLEMENTATION OF THE MESSAGE PASSING INTERFACE ON RTEMS

Arindam Saha
Associate Professor
Department of Electrical & Computer Engineering
Mississippi State University

Abstract

The Message Passing Interface (MPI) is implemented with a real-time operating system (RTOS) called the Real-Time Executive for Multiprocessor Systems (RTEMS). The implementation is based on the shared memory driver based MPCI (Multiprocessor Communications Interface) layer of RTEMS and the public domain portable MPICH code. A RTEMS-specific new device is created for the MPICH implementation by replacing everything below and including the p4 layer with RTEMS-specific message passing code. The basic object used in RTEMS message passing is the message queue. Message queues are distributed in all *N* nodes with *N-1* queues per node such that a sender node can deposit its message to a dedicated queue at the destination. Our MPI implementation should work for all MPI programs thus providing the complete MPI functionality.

AN IMPLEMENTATION OF THE MESSAGE PASSING INTERFACE ON RTEMS

Arindam Saha

1.0 Introduction

The ever increasing demands of faster computational powers and more and more system resources are being met by parallel processing. But, various factors limit the performance of parallel computers. These overheads include context switching, scheduling, load balancing, and most importantly communication between processors. For parallel processing to become effective one must confront these overheads at the outset [1]. For most applications running on parallel computers the communication cost outweighs the computational cost, and while not much can be done to improve the computational powers of a given parallel computer, it is imperative to reduce the communication burden as much as possible. One of the most commonly used parallel processing paradigms is based on message passing between concurrently executing tasks with explicit sending and receiving of messages. Although message passing is traditionally used in distributed memory parallel computers, one can exploit the shared memory capability, if present, to enhance the performance of message passing.

MPI [2] is a message passing library specification that provides a portable and efficient interprocess communication (IPC) mechanism for parallel computers. MPI guarantees reliable in-order data transfer without making any architecture-specific assumptions. MPI is the defacto standard used on a variety of parallel computers including multiprocessors, multicomputers and workstation clusters.

Real-time systems require not only fast response times but also on-time response. So *when* the results are produced is important. To meet the unique requirements of real-time applications the system must have appropriate features[3].

RTEMS is a lightweight operating system ideal for memory-constrained, primarily embedded, both real-time and otherwise, parallelizabe, and computationally intensive applications. It is available free on-line at `http://www.lancelot.gcs.redstone.army.mil/rtems` website. Like many other RTOSs, RTEMS has the following basic features[4]:

- event-driven, priority-based, preemptive scheduling
- multitasking capabilities
- rate monotonic scheduling at user discretion
- priority inheritance (to reduce priority inversion)

- fast interrupt management
- dynamic memory allocation
- high degree of user configurability

and most importantly for this present work

- support for intertask communication and synchronization.

Rest of this report is organized in six sections. In section 2 we discuss the RTEMS details necessary for our work. The MPICH implementation of MPI is discussed in section 3. Our specific implementation is reported in section 4. We draw some conclusions and topics for future work in section 5. Acknowledgments and references are listed in sections 6 and 7 respectively.

2.0 RTEMS Support for Multiprocessors

The two executives used heavily in this work are the *message manager* and the *multiprocessing manager*[4].

2.1 Message Manager

The message manager is a real-time executive that provides communication and synchronization capabilities based on IPC objects called message queues. Message queues can be local or global, created, deleted, flushed or identified, and facilitate message passing by sending, receiving and broadcasting messages. The eight directives provided by the message manager are:

1. `rtems_message_queue_create(`

```
rtems_name name,

rtems_unsigned32 count,

rtems_unsigned32 max_message_size,

rtems_attribute attribute_set,

rtems_id *id

);
```

It creates a message queue residing on the local node with the specified *name* and memory allocated for specified *count* of messages, where each message can be up to *max_message_size* bytes in length. The set of attributes determine whether tasks wait by FIFO or by priority, and whether the queue is local or global. A global queue requires extra overhead but allows the interaction of remote tasks. The total number of global

objects is limited by the *maximum_global_objects* field in the configuration table. This directive returns a message queue *id* which is used to access the queue.

This directive, like most others, returns a status code (refer to the file `..../c/src/exec/support/status.h` file for a full description of all the status codes) which can be RTEMS_SUCCESSFUL (queue created successfully), RTEMS_INVALID_NAME (invalid task name), RTEMS_INVALID-NUMBER (invalid message count0, RTEMS_INVALID_SIZE (invalid message size), RTEMS_TOO_MANY (too many queues created), RTEMS_MP_NOT_CONFIGURED (multiprocessing not configured), or RTEMS_TOO_MANY (too many global objects).

2. `rtems_message_queue_ident(`

rtems_name name,

rtems_unsigned32 node,

rtems_id *id

);

One can use this function to obtain the message queue *id* associated with the queue *name*. A nice feature is the ability to search all the nodes by setting the *node* argument to RTEMS_SEARCH_ALL_NODES. If *node* is a valid node number other than the local node, then only the designated node is searched.

The status code returned by this function can be RTEMS_SUCCESSFUL (queue identified successfully), RTEMS_INVALID_NAME (queue name not found), or RTEMS_INVALID_NODE (invalid node id).

3. `rtems_message_queue_delete(`

```
rtems_id id

);
```

This function deletes the message queue specified by *id*, and although the calling task need not have created the queue it must reside on the same node as the queue. All tasks waiting to receive a message from this deleted queue will be readied and returned a status code indicating the deletion of the queue. If no tasks are waiting and the deleted queue contains messages, then those message buffers are returned to the system pool.

The status code returned by this directive includes RTEMS_SUCCESSFUL (queue deleted successfully), RTEMS_INVALID_ID (invalid queue id), or RTEMS_ILLEGAL_ON_REMOTE_OBJECT (cannot delete remote queue).

4. `rtems_message_queue_send(`

rtems_id id,

void *buffer,

rtems_unsigned32 size

);

This is one of the three directives (the others being `rtems_message_queue_urgent` and `rtems_message_queue_broadcast`) provided by RTEMS to send messages. This procedure enables us to send a message *buffer* of *size* bytes to the message queue denoted by *id*. If the message queue is global and does not reside on the local node then a request is generated to that node to post the message on the specified queue. If a task is waiting at the queue then the message is copied to it's buffer and the task is unblocked. Otherwise, the message is copied to a buffer obtained from the buffer pool. The message buffer is then placed at the rear of the queue. This directive cannot successfully send a message to a message queue which is already full with pending messages.

5. `rtems_message_queue_urgent(`

```
rtems_id id,

void *buffer,

rtems_unsigned32 size

);
```

This directive is identical to `rtems_message_queue_send` when tasks are waiting to receive a message. When no tasks are waiting at the queue, this directive places the message at the front of the queue, whereas the `rtems_message_queue_send` directive places it at the rear.

The status code returned by this procedure can be RTEMS-SUCCESSFUL (message sent successfully), RTEMS_INVALID_ID (invalid queue id), RTEMS_INVALID_SIZE (invalid message size), RTEMS_UNSATISFIED (out of message buffers), or RTEMS_TOO_MANY (queue's limit has been reached).

6. `rtems_message_queue_broadcast(`

```
rtems_id id,

void *buffer,

rtems_unsigned32 size,

rtems_unsigned32 *count

);
```

This function broadcasts as an *atomic* operation. It causes all tasks waiting at the queue specified by *id* to be unblocked and sent the message contained in *buffer*. Before a task is

unblocked, the message *buffer* of *size* bytes is copied to that task's message buffer. The function returns the number of tasks unblocked via *count*. This function is more efficient than the equivalent number of invocations of `rtems_message_queue_send`, thus providing an avenue to improve the performance of MPI collective operations.

The status code returned by this function is RTEMS_SUCCESSFUL (broadcast successful), RTEMS_INVALID_ID (invalid queue id), or RTEMS_INVALID_SIZE (invalid message size).

7. `rtems_message_queue_receive(`

```
rtems_id id,

void *buffer,

rtems_unsigned32 *size,

rtems_unsigned32 option_set,

rtems_interval timeout

);
```

This directive attempts to retrieve a message from the queue specified by *id*. If at least one message is in the queue, then it is copied to *buffer* and the length of message in bytes returned via *size*. On the other hand, if the queue is empty then one of the following situations applies:

- By default, calling task will wait forever for the message to arrive.
- If NO_WAIT option is specified, then the directives returns immediately with an error status code.
- If the *timeout* parameter is set to NO_TIMEOUT then the calling task will wait forever.

The status code returned by this directive can be any one of RTEMS_SUCCESSFUL (message received successfully), RTEMS_INVALID_ID (invalid queue id), RTEMS_UNSATISFIED (queue is empty), RTEMS_TIMEOUT (timed out waiting for message), or RTEMS_OBJECT_WAS_DELETED (queue deleted while waiting).

8. `rtems_message_queue_flush(`

```
rtems_id id,

rtems_unsigned32 *count

);
```

This directive removes all pending messages from the message queue denoted by *id* and returns the number of messages removed via *count*.

The status returned may be either RTEMS_SUCCESSFUL (messages removed successfully) or RTEMS_INVALID_ID (invalid queue id).

2.2 Multiprocessor Manager

RTEMS presents the application a logical view of the multiprocessor system where the boundaries between processors are transparent, thus objects like message queues, tasks, signals, events, semaphores and memory blocks can be designated global and can then be accessed from anywhere in the system. So RTEMS allows the application to view the physically distributed multiprocessor as a virtual single system.

The Multiprocessor Communications Interface (MPCI) layer is a set of user-provided functions that accomplish interprocessor communication. One can visualize this as a thin layer between RTEMS and the underlying hardware to be used by RTEMS during the preparation and processing of remote requests (like sending a message to a global message queue not resident in the caller node). The basic MPCI resources are buffers called packets which are encapsulated within envelopes that contain control information. The addresses of the MPCI routines are placed in a MPCI table. The five MPCI routines are:

1. MPCI_initialization

Prototype: `rtems_mpci_entry user_mpci_initialization(`

```
rtems_configuration_table *configuration
);
```

where *configuration* is the address of user's *Configuration Table*.

This routine is part of the overall *initialize_executive* and must be invoked before any operation on global objects is performed. The primary function of this routine is to create and initialize a pool of packet buffers. This function is implemented by Shm_initialization

2. MPCI_get_packet

Prototype: `rtems_mpci_entry user_mpci_get_packet(`

```
rtems_packet_prefix **packet
);
```

where *packet* is the address of a pointer to a packet. Upon successful return *packet* will contain the address of a packet.

3. MPCI_return_packet

Prototype: `rtems_mpci_entry user_mpci_return_packet(`

```
rtems_packet_prefix *packet
```

```
);
```

where *packet* is the address of a packet. This routine releases a packet to the free packet buffer pool.

4. MPCI_receive_packet

Prototype: `rtems_mpci_entry user_mpci_receive_packet(`

```
rtems_packet_prefix **packet

);
```

where *packet* is a pointer to the address of a packet. This routine obtains a previously arrived packet. If a message is available, then *packet* will contain the address of the message from another node, otherwise contain NULL.

5. MPCI_send_packet

Prototype: `rtems_mpci_entry user_mpci_send_packet(`

rtems_unsigned32 node,

```
rtems_packet_prefix **packet

rtems_unsigned32 packet_length

);
```

where *node* is the destination node number, *packet* is the address of a packet containing a message, and *packet_length* is the number of bytes in the message. This routine accomplishes the basic function of sending a message from one node to another. The *node* parameter should be set to zero to achieve a broadcast. The *packet_length* may be used to avoid sending unnecessary data.

2.2.1 The Shared Memory Driver for MPCI in RTEMS

We have used the shared memory driver included in RTEMS3.6.0 for the implementation of the MPCI layer. This driver is located in the `src/lib/libbsp/shmdr` directory. All MPCI functions are implemented with this shared memory driver that assumes the presence of a shared memory for all non-local operations. Semaphores are duly used to prevent contention during the access of shared resources like global message queues. All messages are packed in simple packet structures

3.0 Review of the MPICH Implementation

MPICH is a portable implementation of MPI (jointly developed by the Argonne National Laboratory and Mississippi State University [5]) that allows quick porting to new plat-

forms. The software abstraction of MPICH has three layers - the machine-independent message passing API called MPIR, an Abstract Device Interface (ADI) [6] that defines a conceptual set of services which should be provided by any lower level communication mechanism for the MPIR layer, and the machine-dependent Device layer which is basically the implementation of the ADI services on a given platform. One can choose any one of these three layers and implement everything below it to port MPICH to a new target platform. The revised ADI design (called ADI-2 [7-8]) addresses the issues connected with the use of multiple communication devices and heterogeneous communication protocols. To facilitate porting MPICH provides a Channel Device Interface (CDI) that conceptualizes the services of the device layer and consists of a set of few functions that need to be implemented on the target platform [9]. In this work we have chosen the path of implementing the CDI. The rest of this section discusses the CDI.

There are a "bare minimum" five functions that constitute the CDI. *All* MPI functions are converted to these five primitive data exchange functions. The message is sent in two parts. The *control part* contains information like data size, source, destination, message tag, communicator, etc., and the *data part* contains the actual data. If the data is small enough (i.e. less than a threshold size to be set by the implementor) then it is sent with the control part. The amount of data in the control part can be set by the command line option `-pkt_size` option and is typically set to 1024 bytes. These five CDI functions are as follows:

1. `int MPID_ControlMsgAvail(`

```
void

);
```

This function returns a boolean value indicating whether a control packet from *any* source is available.

2. `void MPID_RecvAnyControl(`

```
MPID_PKT_T *pkt,

int size,

int *from

);
```

This reads the next control message of length *size* into the buffer pointed to by *pkt* and returns the rank of the source in the in-out parameter ***from*. This is a blocking function. The control packet is defined in the `packet.h` file.

3. `void MPID_SendControl(`

```
MPID_PKT_T *pkt,

int size,
```

```
int dest

);
```

This function sends a control packet of length *size* pointed to by *pkt* to the destination process with rank *dest*. This is a non-blocking function, returns even if no data is sent.

4. `void MPID_RecvFromChannel(`

```
void *buf,

int maxsize,

int from

);
```

This function receives the data part of the original message and the word *channel* here refers to the source id as specified by the *from* parameter. The received message is copied into *buf* and *maxsize* refers to the maximum size of data that can be received without truncation. It is assumed that the control packet corresponding to this data packet has already been received and that *maxsize* and *from* are known. This function blocks until the entire packet is received.

5. `void MPID_SendChannel(`

```
void *buf,

int size,

int dest

);
```

After the corresponding control packet has been sent, this function is used to send the data part contained in *buf* of length *size* to the destination process *dest*. This function, like `MPID_SendControl`, is non-blocking.

The issue of progress order is controversial. MPI guarantees that the delivery is in-order between any two processes, but the order is undefined when two or more processes send to one other process. From the implementation standpoint, this means that for a long (control + data) message the sending process performs `MPID_SendChannel` after the `MPID_SendControl` for that message and before performing a `MPID_SendControl` for any other message.

For efficiency purposes, in order to not require that the control message may be delivered without requiring a matching receive, the optional `MPID_SendControlBlock` function may be used. In this implementation, `MPID_SendControl` is used instead. Also, the CDI provides optional functions for non-blocking data transfer by allowing some data transfers to be left uncompleted until they can be received at their destination. In our

implementation we have not taken advantage of these functions and have thus have *defined* the macros `PI_NO_NRECV` and `PI_NO_NSEND`.

There are two data exchange mechanisms - *eager* and *rendezvous*. In the eager protocol, data is sent to the destination immediately and if the destination has not posted a MPI_Recv for it (i.e, the data is unexpected), the receiver must allocate space to store the data locally. This, although may cause memory problems in the case of really long messages, offers the best performance. In the rendezvous protocol, data is sent only when requested. when posting a matching receive the destination process request the source for data. This is more robust but generally less efficient. Unfortunately, in MPICH one of these two protocols must be used exclusively.

MPI caters to both interrupt-driven and status polling implementations. To provide an *immediate response* to an `MPI_Recv` an interrupt is required and this is the default in MPICH. But one can modify the `MPID_CHECK_DEVICE` macro to poll a device on receive.

4.0 Our Implementation

Our implementation is based on MPICH. Essentially, we have replaced everything below and including the p4 layer of MPICH with RTEMS-specific code. The five CDI functions listed in Section 3 are implemented in RTEMS.

4.1 Carving Out MPICH for Our Device

Our implementation is heavily based on the MPICH implementation - specifically the *ch2* device interface. We have retained the entire `src` directory minus the *mpe* and debug code. We have put all the include files in one directory. From the `mpid` directory we have taken the `ch2` and `util` directories. We then ran the `NewDevice` script to create the header files (which are simply meant for Intel NX) as our starting point. Following is the list of files modified/created:

- `channel.h`
- `chdef.h`
- `packets.h`
- `rtemspriv.c`

The implementation details are discussed in Section 4.4.

4.2 Installing the RTEMS3.6.0 UNIX port

The installation of the RTEMS UNIX port turned out to be a non-trivial task. RTEMS 3.6.0 is downloaded from the `lancelot.gcs.redstone.army.mil` website. From this same site we also downloaded the NEWLIB copy with RTEMS awareness. By executing the `configure` script from the top level of the NEWLIB source tree it is config-

ured for the *unix-solaris-rtems* target. A word to the wise - one has to be careful in avoiding conflicts between the native system files and NEWLIB files.

Now RTEMS is installed as follows:

1. Environment variables are set in the `c/Modules/rtems/oar-solaris2` file and *sourced.*
2. There are three Makefiles that are modified according to our needs - `make/custom/solaris2.cfg`, `make/compilers/gcc-solaris2.cfg`, and `make/os/Solaris-2.3.cfg`. To increase the maximum packet size to 16kbytes we have put `DEFINES += -DMAX_ENVELOPE_SIZE=0x4000` in the `gcc-solaris2.cfg` file. In that same file we made provision to link the two MPI-specific libraries by defining the LINK_FILES variable appropriately (we have included these two libraries twice to satisfy cross dependencies!). Since we don't have (and don't need) any `rtems.o` file we have to remove it from being included in the `solaris2.cfg` file.
3. We now run `make install` from the top level of the RTEMS source tree. To conform to our shell we have to replace bash with ksh in the files `install-if-change` and `rtems-glom`. The file `cpu.c` cannot be compiled due to *libc* conflicts between the native and NEWLIB files. So we compile `cpu.c` manually and separately by removing the NEWLIB include directory. since we have increased the maximum packet size we have correspondingly increased the total amount of shared memory size by including `-DRTEMS_SHM_SIZE=1048576` while compiling the `cpu.c` file.
4. Finally, to run RTEMS-MPI programs we have modified the `runtest` script.

4.3 Modifying RTEMS

The big advantage of using RTEMS is that with the available source code and the well structured modular nature of the software one can enhance RTEMS to provide added functionality specific to one's needs. For example, we need a function to *peek* at message queues to find out whether a message is available for receiving. With the existing RTEMS, one can do this by doing a `rtems_message_queue_receive` which will return a status code indicating empty queue if that is the case. But an unwanted message will be received in this process. We added the directive `rtems_message_queue_inquire` to RTEMS to return the number (including zero) of pending messages in a message queue. The prototype is as follows:

```
rtems_message_queue_inquire(

rtems_id id,

rtems_unsigned32 *count

);
```

This directive finds out the total number of pending messages in the message queue denoted by *id* and returns that number via *count*. The function returns a zero if the queue is

empty. The message queue is unchanged. The status returned may be either RTEMS_SUCCESSFUL (inquiry successful) or RTEMS_INVALID_ID (invalid queue id). We now list the procedure we followed to insert this new function so that future enhancements can be made when necessary. First we looked at the `msg.c` file (the message manager) to find out that among the existing functions, `rtems_message_queue_flush` requires the least amount of change to derive our new function. Then, we checked the `coremsg.h` file to find out there exists a field called `number_of_pending_messages` in the `message_queue` data structure. So we put the prototype of our new function in the `message.h` file, and wrote simple code for this function in the `msg.c` file. The actual count is computed as

```
*count = the_message_queue->message_queue.number_of_
         pending_messages.
```

We then added two values

```
RTEMS_MESSAGE_QUEUE_INQUIRE_REQUEST=13 and
RTEMS_MESSAGE_QUEUE_INQUIRE_RESPONSE=14
```

in the `Message_queue_MP_Remote_operations` *type* in the `msgmp.h` file. Finally we modified the `msgmp.c` file to take care of the extra two *cases*.

4.4 Implementation Details

We have implemented the five channel interface functions within RTEMS without using any UNIX system calls. The message queue is the fundamental object used to perform message passing within RTEMS. Every node has *N-1* message queues such that there is a dedicated queue for each one of the remaining *N-1* nodes. During a send operation the source node transfers the message to *message_queue[destination][source]*. During a receive operation the receiving node receives from *any* non-empty message queue residing in that node. The overhead per message queue is only about 150bytes for the control block and an extra 50bytes if the queue is global. Thus the overhead for having distributed message queues is rather small and manageable even for space-constrained systems. We do not use any other queues at RTEMS level. The `MPID_PKT_T` packet structure is used for send/receive of control packets. For send/receive of *channel* data we simply use `void *buffer`. We have taken special care to interface the ADI layer of MPICH and our low-level message passing layer in RTEMS. Some features are easy to change while others may take some effort. Users can quickly change the maximum message length or the maximum number of messages per message queue. But if a change is made in the maximum packet size or the total shared memory size within the MPCI layer of RTEMS then RTEMS has to be *cleaned* and recompiled. All the five CDI functions are coded in the `rtemspriv.c` file. The interface between MPICH ADI and RTEMS is done via the `channel.h` and `chdef.h files`. Among the possible protocols, we have chosen the *eager* protocol for our implementation. The *rendezvous* protocol may be used with

slight modification. We have simply used the RTEMS function called `Multiprocessing_configuration.node` to return a node's rank.

We have ensured that major MPI functions perform correctly. Due to lack of time we did not get a chance to evaluate all MPI functions. We emphasize that since the five functions of CDI are sufficient for the entire MPIC in MPICH there is no reason why all the untested MPI functions should not work correctly. There is always a room for optimization. Some of the possible future optimizations are discussed in section 5. We conclude this section with a detailed discussion of the two most important and most commonly used functions in MPI.

Among MPI users, the blocking send and blocking receive are heavily used. The flow-chart of a Blocking Send (the RTEMS part is magnified in inset) is shown in Figure 1.

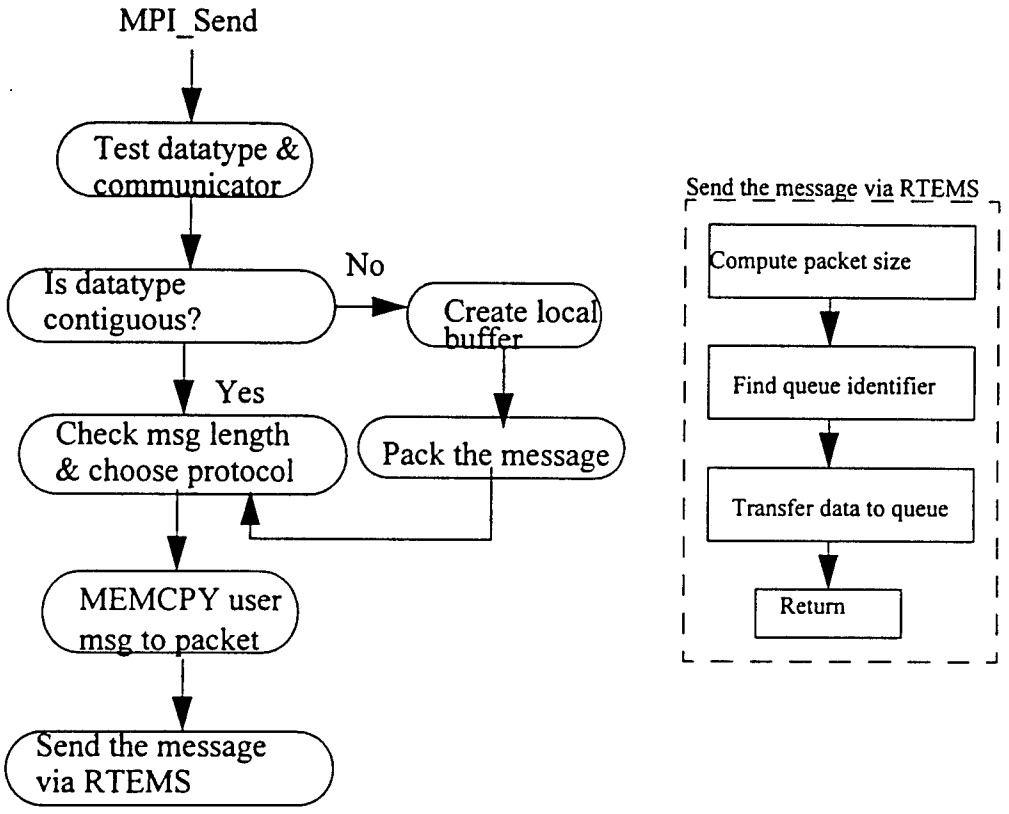


Figure 1. Blocking Send in MPI

The flowchart of a Blocking Receive (the RTEMS part is magnified in inset) is shown in Figure 2.

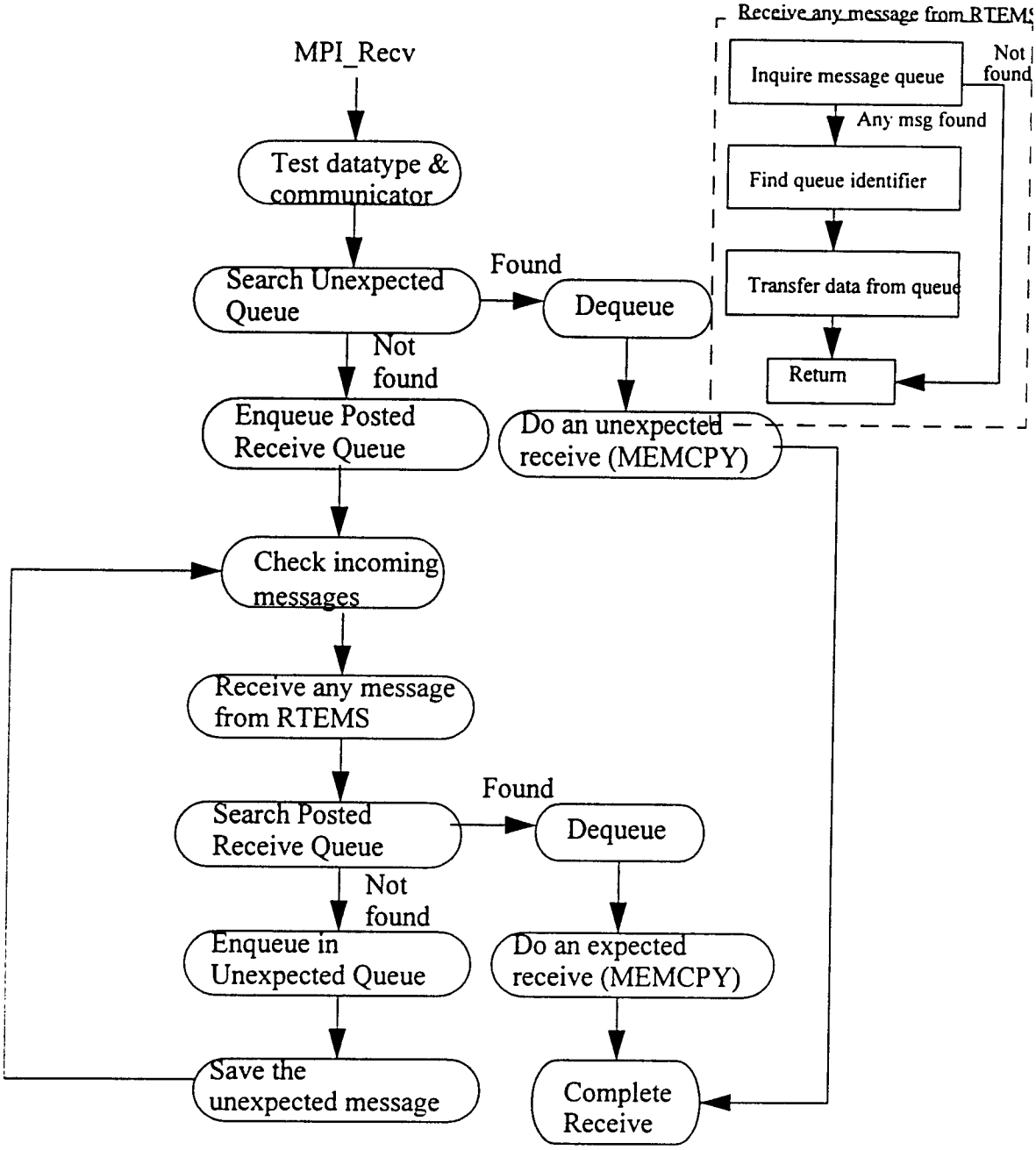


Figure 2. Blocking Receive in MPI

4.5 Test Results

We have tested our MPI implementation with a few standard MPI programs. We have not done any performance testing and have only limited our tests to functionality of MPI. All our tests were executed successfully. Thus this is a *complete* implementation providing all MPI functions to the user. We have tested MPI programs with the following functionality

- blocking send and receive
- asynchronous send and receive
- collective communication

Following is a suggested way of running MPI programs under RTEMS (assuming you are in `rtems-3.6.0/c/src/tests/mptests` directory and running on 4 nodes):

- Create your own directory, say `mympi` (`mkdir mympi`)
- You should create 4 directories - `node1, node2, node3,` and `node4` - with the appropriate `Makefile` in each one. For the sake of convenience, copy these directories from an existing mpi test directory (say `cp -r ../mpi1/node? .`). Change all four Makefiles by putting appropriate test name. Also copy the `Makefile, init.c, system.h` and `mpi.h` from that same test directory.
- Now create and edit your own MPI program, say `mainprog.c`. This should be a RTEMS task with name `Init_task`.
- Clean files by doing a `make clobber`. Then do a `make`. Fix all errors. After a successful make, you should now have an executable file (with `.exe` extension) in each one of the four node directories.
- Copy the four executable to the `c/solaris2/tests` directory (if not already done by the `Makefile`)
- Run the programs with `runtest mympi-node1.exe` command. Output is logged in four files in the `c/solaris2/tests/log` directory.

5.0 Discussions

MPI is implemented with the RTEMS RTOS. The implementation is based on the shared memory driver based MPC layer of RTEMS and the public domain portable MPICH code. A RTEMS-specific new device is created for the MPICH implementation by replacing everything below and including the p4 layer with RTEMS-specific message passing code. The basic object used in RTEMS message passing is the message queue. Message queues are distributed in all *N* nodes with *N-1* queues per node such that a sender node can deposit its message to a dedicated queue at the destination. Our MPI implementation should work for all MPI programs thus providing the complete MPI functionality.

The following is a list of open issues that should be addressed in future work:

- Performance testing should be done and efforts should be made to reduce message passing latency.
- Only the essential part of MPICH should be retained. The rest should be methodically removed.
- MPI is evolving. We have used MPI1.0 in this work. It is worthwhile to investigate both embedded MPI and real-time MPI as and when they become ready. The added functionality of the recently introduced MPI2.0 should also be looked into.

- The number of message queues can be reduced to one per node with an additional appropriate local queue per node to handle unwanted receives. This may reduce the space requirements although it unclear by how much because the extra overhead for a global queue is less than 100 bytes.
- The shared memory driver based MPCI layer will have to be redone for distributed memory parallel machines as well for machines with very little shared memory.
- Both the initialization and finalization parts of MPI need further optimization.
- This work is done with version 3.6.0 of RTEMS - the most current publicly available version. Migration to future versions of RTEMS should be done especially since version 4.0.0 is imminent.
- Currently we are communicating the entire `MPID_PKT_T` packet structure in RTEMS. One can easily map this packet to a simpler structure, retaining only the essentials, that will reduce the space requirements per message queue.
- During a receive, we get *any* message rather than a specific one from a speicific source or with other specific characteristics. The unwanted receives are handled appropriately by the MPICH code. One can modify the message queue receive or add a separate receive in RTEMS such that specific messages can be received.
- Some relevant questions need to answered - whether RTEMS statically allocates the message queue space or not, what the minimun control information size is per packet, how exactly does RTEMS perform non-local message queue operations and how they may be improved.

6.0 Acknowledgments

We wish to thank four people in particular for their help, suggestions, and discussions - Mark Linderman of Rome Laboratory, Dennis Fitzgerald of Kaman Sciences, William Gropp of Argonne National Laboratory, and Joel Sherrill of On-Line Applications Research Corporation.

7.0 References

1.Saha, A., *The Design and Analysis of Parallel Recursive Algorithms in the Presence of Overheads*, Ph.D. dissertation, Lehigh University, 1991.

2. The MPI Forum, *The Message-Passing Interface Standard*, `http://www.mcs.anl.gov/mpi/mpi-report/mpi-report.html`, 1995.

3. Saha, A., *Evaluation of Network Routers for Real-time Parallel Computers*, Technical Report, Wright Laboratory, Wright-Patterson Air Force Base, 1994.

4. U.S. Army Missile Command, Redstone Arsenal, Alabama, and OAR Corporation, *RTEMS User Manual*, version 3.6.0, 1996.

5. Gropp, W., Lusk, E., and Skejellum, A., *MPICH: A High-Performance Portable Implementation for the Message-Passing Interface*, ftp://ftp.mcs.anl.gov/pub/mpi/mpicharticle.ps..

6.Gropp, W. and Lusk, E., *MPICH ADI Implementation Reference Manual*, ftp://ftp.mcs.anl.gov/pub/mpi/adiman.ps.

7. Gropp, W., and Lusk, E., *MPICH Working Note: The Second-Generation ADI for the MPICH Implementation of MPI*, ftp://ftp.mcs.anl.gov/pub/mpi/workingnote/nextgen.ps.

8. Gropp, W., and Lusk, E., *MPICH Working Note: The Implementation of the Second-Generation MPICH ADI*, ftp://ftp.mcs.anl.gov/pub/mpi/workingnote/adi2imp.ps.

9. Gropp, W., and Lusk, E., *MPICH Working Note: Creating a New MPICH Device Using the Channel Interface (DRAFT)*, ftp://ftp.mcs.anl.gov/pub/mpi/workingnote/newadi.ps.

A STUDY OF INTEGRATED AND INTELLIGENT NETWORK MANAGEMENT

Ravi Sankar
Associate Professor
Department of Electrical Engineering

University of South Florida
4202 East Fowler Avenue
Tampa, Florida 33620-5350

Final Report for:
Summer Faculty Research Program
Rome Laboratory

Sponsored by:
Air Force Office of Scientific Research
Bolling AFB, Washington, DC

and

Rome Laboratory
Rome, NY 13441

August 1997

A STUDY OF INTEGRATED AND INTELLIGENT NETWORK MANAGEMENT

Ravi Sankar
Associate Professor
Department of Electrical Engineering
University of South Florida

Abstract

The development of an integrated and intelligent network management platform for the Information for the Warrior (IFTW) program was studied. Specifically, the issues in performance and resource management are outlined in this report. The performance trending technique for predicting *soft* network failures was investigated and the concept is described here. The feasibility of applying fuzzy rule-based system for hand-over between networks before any loss of communication was studied and the evaluation of the fuzzy system is reported. Also, a predictor-based architecture developed for a dynamic bandwidth management in an ATM environment is described.

A STUDY OF INTEGRATED AND INTELLIGENT NETWORK MANAGEMENT

Ravi Sankar

1. Introduction

This final report summarizes the research work performed under the summer faculty research program sponsored by the Air Force Office of Scientific Research (AFOSR) and Rome Laboratory.

2. Statement of the Problem

The objectives of the Information for the Warrior (IFTW) program are to develop and demonstrate technologies for global access to C4I information. The program focuses on integrating advanced technologies in the areas of network management, network communications (ATM, RF and Satellite communications), and real-time information management.

In the IFTW program, the communication is expected over a wide range of environments including wireless or space environment (HF, UHF and Satellite communication) and wireline or terrestrial environment (ATM network communication). The characteristics of wireless channels are quite different from that of wired ATM network and the problems encountered for operation and management of these networks are also diverse. For instance, wireless channels are noisy (high bit error rate), limited in bandwidth or capacity (low data rate), etc., while the wired channels have just the opposite characteristics. Even within the wireless channels, the characteristics may differ if it is a HF or a satellite link. HF links are quite unpredictable due to the effects of multipath and fading while satellite links are associated with large propagation delays. Further the problem is exacerbated if the communication is across multiple networks (communication environment consists of a combination of various channels). The quality of service (QoS) delivery and network performance would now be constrained by the weakest link in the chain.

Satisfying QoS requirements for traffic generated by different services, while maintaining a high utilization of network resources (e.g., bandwidth and buffers) is the objective of efficient resource management of broadband networks. Since the network has to integrate a wide range of services/applications with diverse traffic characteristics, QoS requirements and performance constraints, the resource allocation becomes very complex and difficult but is critical to the success of ATM-based technology deployment [1-2].

In such a diverse and complex networking environment, the objective of the network management system is to dynamically configure the best available channel according to the requirements (QoS specification, priority, etc.,). Further the adaptive system should have the ability to sense the communication environment, and change the network operation to improve the performance.

This research work addresses the issue of network resource management to efficiently allocate resources (bandwidth) for the best path between the end nodes once the connectivity is established. It also addresses the issue

of performance management for selecting the network of choice for the initial connection using performance trending technique and fuzzy control for hand-over between networks.

3. Project Objectives

For the network management, the objective is to develop an adaptive system to make automated intelligent decisions on configuring communication assets, optimize network resources such as bandwidth and dynamically adapt to the network and user requirements. The main objective of this research is to study the feasibility of developing an integrated and intelligent network management platform to efficiently allocate resources (bandwidth) and to optimally select the network of choice for the initial connection and hand-over between networks using performance trending technique and fuzzy control. The tasks addressed in this report include:

- Developing performance trending technique for network management.
- Studying the feasibility of applying fuzzy control for hand-over between networks.
- Examining video traffic related management issues.
- Developing an integrated framework for resource management of video traffic using prediction and short-term control.

4. Performance Trending Technique for Network Management: Concept and Methodology

Network management addresses the issue of efficient use of network resources while providing reliable service to the user. Four basic areas of management include: configuration and control administration, security, and performance management [3-4]. The network performance management can be further categorized as fault management: process to recover a network from system failures and performance maintenance: process to recover the network from failure to maintain the performance level, i.e., specified level of QoS or grade of service (GoS). These are termed as *hard* and *soft* failures, respectively [5]. An important difference between the two is in the network (or resource) availability. In the case of hard failures, the network will be down unless it is fault tolerant while in the case of soft failures, the network can still be operating at a lower level of QoS. What is of interest here is the failures that occur by the gradual degradation of performance. The end-to-end delay provides an indication of the network congestion while bit error rate provides an indication of the reliability of the link. By real-time monitoring, analyzing, and performance trending these network parameters, an early detection of failure is possible. The network recovery is by a hand-over to a target network before any loss of communication while dynamically optimizing network bandwidth in an ATM environment [6].

The performance trending involves non-linear sampled monitoring of network parameters and using short- and long-term statistical analysis to detect the early onset of failure [5]. This technique is also used to track the network resources and provide information to help choose the network for initial connection. An important aspect of trending is the measurement frequency. The network QoS parameters are specified always as a range with a minimum and a maximum value (e.g., average delay: $[d_{min}, d_{max}]$ and bandwidth: $[bw_{min}, bw_{max}]$). For example--in

the case of delay when the short-term average of the measured values is near or below the lower value (d_{min}) the sampling rate is normal. If it gets near or above the detection threshold which is set slightly below the maximum value (d_{max}), then the measurement frequency is progressively increased so as to detect the onset of failure (violation of QoS contract) as shown in Figure 1. At this time, the communication is handed over to an alternative network which satisfies the user specified criteria. It is very difficult to obtain an analytical model combining all the network performance parameters to find the optimal hand-over point. Fuzzy logic rule-based systems have demonstrated the ability to make intelligent decisions using soft thresholds [7-8].

5. Optimal Selection and Hand-over between Networks using Fuzzy Control

Fuzzy set theory [9-10] allows a linguistic representation of the control and operational laws of a system where accurate analytical models either are not available or impractical. The main strength of fuzzy set theory is that it excels in dealing with imprecision. Fuzzy set allows partial membership. It has a gradual transition from full membership of a set to full non-membership (though not simultaneously). Thus fuzzy set, a generalization of classical set, allows to express or represent imprecise information qualitatively by using fuzzy terms. The members of the fuzzy set 'A' are assigned values specifying the degree of membership according to the membership function denoted as μ_A, i.e., $\mu_A : X \rightarrow [0,1]$.

The block diagram of a typical fuzzy system is shown in Figure 2. The input and output of the fuzzy system are $\mathbf{x} \in \boldsymbol{R}^n$ and $\mathbf{y} \in \boldsymbol{R}$, respectively, where $\boldsymbol{R}$ is the set of real numbers. Information to the fuzzy system first enters the *Fuzzifier,* where it is fuzzified. The fuzzified data is passed to the *Inference Engine.* The Inference Engine matches the fuzzified data against a set of *Fuzzy Rules* using fuzzy techniques to produce output fuzzy sets. The outputs of the fuzzy sets are then passed to the defuzzifier which computes a crisp output value by the centroid algorithm. Fuzzy logic uses linguistic variables to map the input fuzzy variables to the output fuzzy variable(s). This is accomplished by using fuzzy **IF-THEN** rules.

The illustrative example presented below shows how two QoS parameters can be utilized to determine the hand-over between two networks. The parameters considered are average delay (delay) and bit-error-rate (BER). The fuzzy membership functions for them are shown in Figures 3-4. Figure 5 shows the membership function for the hand-over factor (HOF) which is the output of the fuzzy system. Fuzzy rules map the inputs (*delay and BER*) to the output (*HOF*) and is of the form:

IF *delay = large and BER = high,* **THEN** *HOF = high*
IF *delay = large and BER = medium,* **THEN** *HOF = high*
IF *delay = large and BER = low,* **THEN** *HOF = high*
IF *delay = medium and BER = high,* **THEN** *HOF = high*
IF *delay = medium and BER = medium,* **THEN** *HOF = medium*
IF *delay = medium and BER = low,* **THEN** *HOF = medium*
IF *delay = small and BER = high,* **THEN** *HOF = high*
IF *delay = small and BER = medium,* **THEN** *HOF = medium*
IF *delay = small and BER = low,* **THEN** *HOF = low*

The fuzzy algorithm will initiate a hand-over from the current network to the target network when

$(HOF)_{current} > (HOF)_{target}$

The fuzzy rule-based control system will be able to incorporate other network parameters such as available bandwidth or capacity, user/application priorities, service types, etc.,

A soft hand-over is made, i.e., make-before-break, to ensure a smooth switching between the network. As long as there is an alternative network that can satisfy the specified QoS criteria, there will be no loss of communication while hand-over. Similar fuzzy-based decision can be used to select a network for the initial connection.

6. Fuzzy System Evaluation and Discussion

Experiments were performed to evaluate the effectiveness of fuzzy control for hand-over between two networks. For the case study, the fuzzy membership functions shown in Figures 3-5 were utilized with delay: d_{min} = 100 ms, d_{max} = 200 ms and BER: $b_{min} = 10^{-10}$, b_{max} 10^{-2}. Table 1 shows the measurement values of the network parameters: delay and BER for the two networks and their hand-over factor (HOF). If Network 1 is the current operating network and Network 2 is the target network, then there will be no hand-over initiated since HOF for Network 1 is less than that of Network 2. The computation of HOF is explained below.

Consider the case of Network 1 with delay = 130 ms and BER = 10^{-7}. The delay maps to degree of memberships in the fuzzy sets *small* and *medium* with μ_{small}(delay=130) = 0.8 and μ_{medium}(delay=130) = 0.2, respectively. Similarly the BER maps to degree of memberships in the fuzzy sets *low* and *medium* with $\mu_{low}(BER=10^{-7}) = 0.5$ and $\mu_{medium}(BER=10^{-7}) = 0.5$. By applying the fuzzy rules, the four combinations of inputs (delay and BER) were mapped to output (HOF) as:

μ_{low}(HOF) = 0.5

μ_{medium}(BER) = 0.2

μ_{medium}(BER) = 0.5

μ_{medium}(BER) = 0.2

The first value μ_{low}(HOF) = 0.5 was obtained by ANDing the memberships for delay and BER, i.e., μ_{small}(delay=130) = 0.8 and $\mu_{low}(BER=10^{-7}) = 0.5$, which is the minimum value in the fuzzy set theory and using the rule: **IF** *delay = small and BER = low,* **THEN** *HOF = low.* Similarly others were obtained and then a crisp output of hand-over factor (HOF) was computed using a centroid algorithm. From Table 1, HOF for Network 1 was 0.41.

Network 2 with delay = 170 ms and BER = 10^{-7} was considered as a target. Similarly the parameters were mapped to degree of memberships in the associated fuzzy sets and then mapped to output. HOF for Network 2 was computed to be 0.68 as shown in the Table.

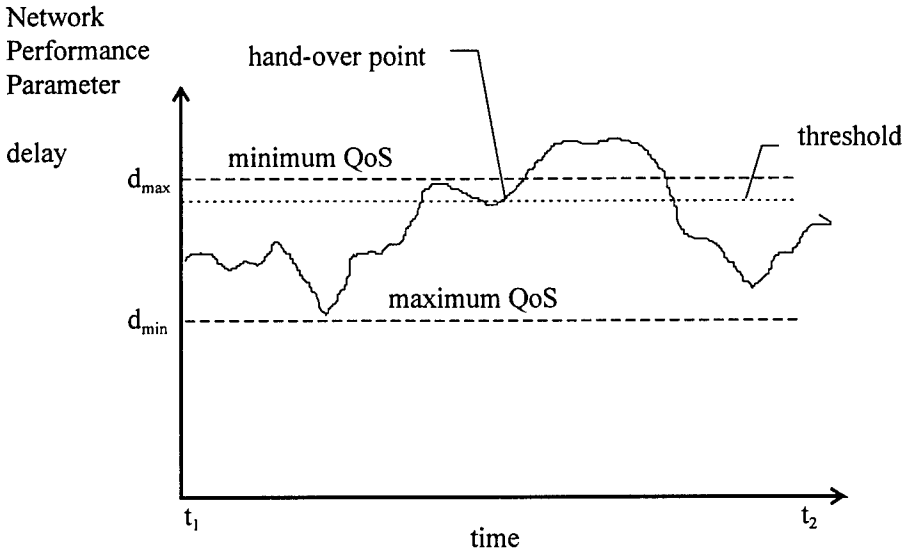


Figure 1. Performance trending illustration

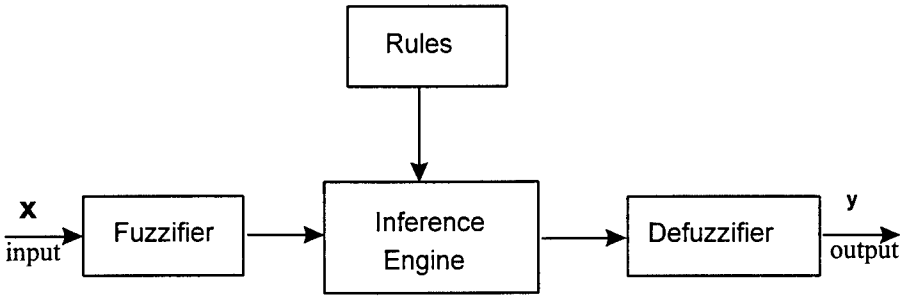


Figure 2. Block diagram of a fuzzy system.

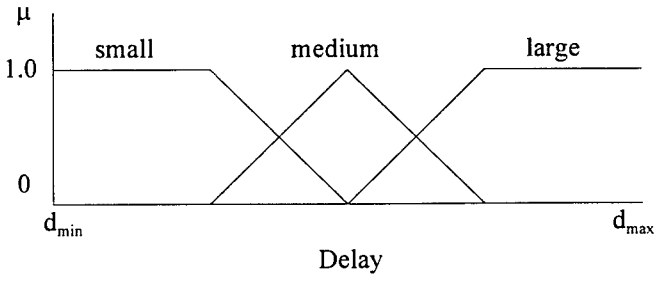


Figure 3. Degree of membership function for delay.

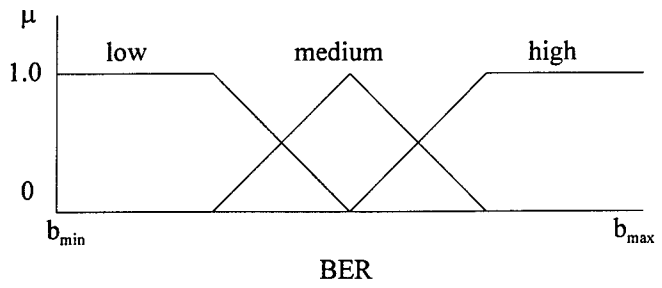


Figure 4. Degree of membership function for bit-error-rate.

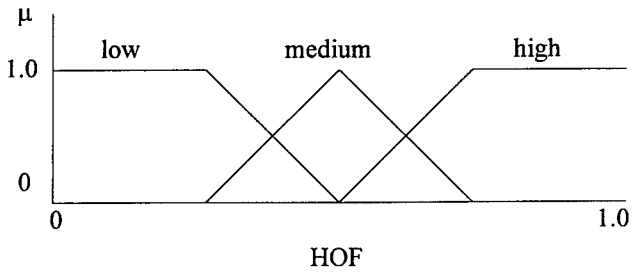


Figure 5. Degree of membership function for hand-over factor.

Table 1. Hand-over factor (HOF) for networks.

Network Performance Parameter	Network 1 (current)	Network 2 (target)
delay (ms)	130	170
BER	10^{-7}	10^{-7}
HOF	0.4107	0.6786

7. Resource Management of Video Traffic in ATM Networks: Background

Asynchronous Transfer Mode (ATM), the standard mode of transmission for Broadband Integrated Service Digital Network (B-ISDN), is envisioned to be the global, integrated, multi-service, fast packet-switched network with provisions for dynamic bandwidth allocation and quality of service (QoS) guarantees. The traffic management issues which include call admission control, policing, scheduling, and congestion control have to be addressed before this vision becomes a reality. Satisfying QoS requirements for traffic generated by different services, while maintaining a high utilization of network resources (e.g., bandwidth and buffers) is the objective of efficient resource management. Since the ATM network has to integrate a broad range of services/ applications with diverse traffic characteristics, QoS requirements and performance constraints, the resource allocation becomes very complex and difficult but is critical to the success of ATM technology deployment.

The statistical multiplexing gain (SMG) obtained must be carefully weighed against the potential problem of degradation of QoS promised to the users at the time of call admission. The ultimate goal is to design a network that can dynamically adapt to the changing needs of QoS of the clients, but still provide guarantees per session (a session is defined as an end-to-end video connection across an ATM network over which two end nodes communicate, a virtual circuit (VC) in ATM networks). The QoS (e.g., delay, cell loss rate, cell delay variation) requirements for a session are most stringent for real-time video traffic that is expected to occupy a large bandwidth of broadband networks.

Thus, novel dynamic bandwidth allocation schemes are necessary in order to efficiently utilize the network resources (e.g., bandwidth, buffers) and maximize the number of video sessions that can be supported with existing resources. This report presents various VBR video traffic related management issues and describes an integrated framework as a solution to the complex problems of traffic and resource management of real-time video services.

8. Issues in Resource Management of Video Traffic

The issues and approaches relevant to the VBR video traffic management can be summarized as follows: The statistical characterization of the VBR video traffic is difficult mainly due to the fact that different video streams exhibit diverse characteristics (e.g., video conferencing that does not exhibit much of scene activity in contrast to a movie that exhibits more scene activity. The aggregation of video streams makes it more challenging to determine the statistical gain that can be obtained across the sessions in order to utilize the resources effectively.
These factors lead to the following:

i) The call-admission control algorithms have to make decisions based on prior statistical characterization of the video traffic that may not be accurate enough. The advance reservation schemes suffer from the same disadvantage, i.e., they have to rely on prior approximate bandwidth estimates.

ii) Online adaptation to the changing video traffic rates is essential due to the delay constraints imposed by real-time VBR video.

iii) Frequent bandwidth re-negotiations are undesirable due to the overhead involved in fast negotiating protocols and the delay involved in the process of negotiation.

iv) Online adaptation to the change of QoS requirements based on end-users preference is highly desirable.

In order to address these issues, an integrated approach needs to be formulated rather than approaching the problem in isolation (e.g., based solely on equivalent bandwidth allocation schemes).

9. Application of Predictor-based Architecture for Dynamic Bandwidth Management

Figure 1 depicts an integrated approach for VBR video traffic management based on traffic prediction that facilitates the online adaptation to the changing traffic rates. The correlation properties of the VBR video traffic make traffic prediction possible and based on the predictor estimates online adaptation to traffic rates can be done easily [6, 11-12]. Based on the traffic estimates for future adaptation intervals, the predictor system dynamically allocates the bandwidth to various ongoing video sessions. The short-term controller (STC) works at cell-level scheduling while the predictor system works at the burst-level (frame).

Figure 2 depicts the predictor-STC system for a dynamic bandwidth management. The purpose of STC is to reduce the effects of prediction errors on the queues of individual sessions and at the same time to exploit the statistical multiplexing gain across the sessions. The total capacity (bandwidth) that is available for the predictor system for the allocation among the sessions is obtained from the call-admission block. During the call-admission phase, the call-admission algorithm relies on prior statistical characterization of the prospective video sessions that may not be accurate enough leading to an approximate estimation of bandwidth requirement. Thus the QoS Change and Bandwidth Re-negotiation (QSCBR) module plays an important role for the necessary re-negotiations for the desired bandwidth that may be needed by the sessions after admission. The Predictor-STC system provides a mechanism for online adaptation to traffic rates of individual sessions and at the same time exploits the statistical gain across the sessions thereby decreasing the need for frequent bandwidth re-negotiations. The change in end-user based picture quality requirements are considered as QoS changes (equivalently translated to the bandwidth requirements) of the ongoing video sessions that are taken care by the QSCBR module. Further research must be carried out to study the decrease in bandwidth re-negotiations that are possible due to the exploitation of statistical gain across the sessions.

Simple adaptive schemes based on online traffic measurements coupled with aggregate traffic based equivalent bandwidth call-admission control mechanisms give rise to better utilization of network resources and efficient transportation of video traffic. The coordination between the modules depicted in Figure 1 viz., Predictor-STC, Call-admission and QSCBR, is very essential for a better QoS adaptation and network resource utilization. In order to achieve realistic performance measurements with respect to real-time traffic, it is essential to evaluate the network in such an integrated framework.

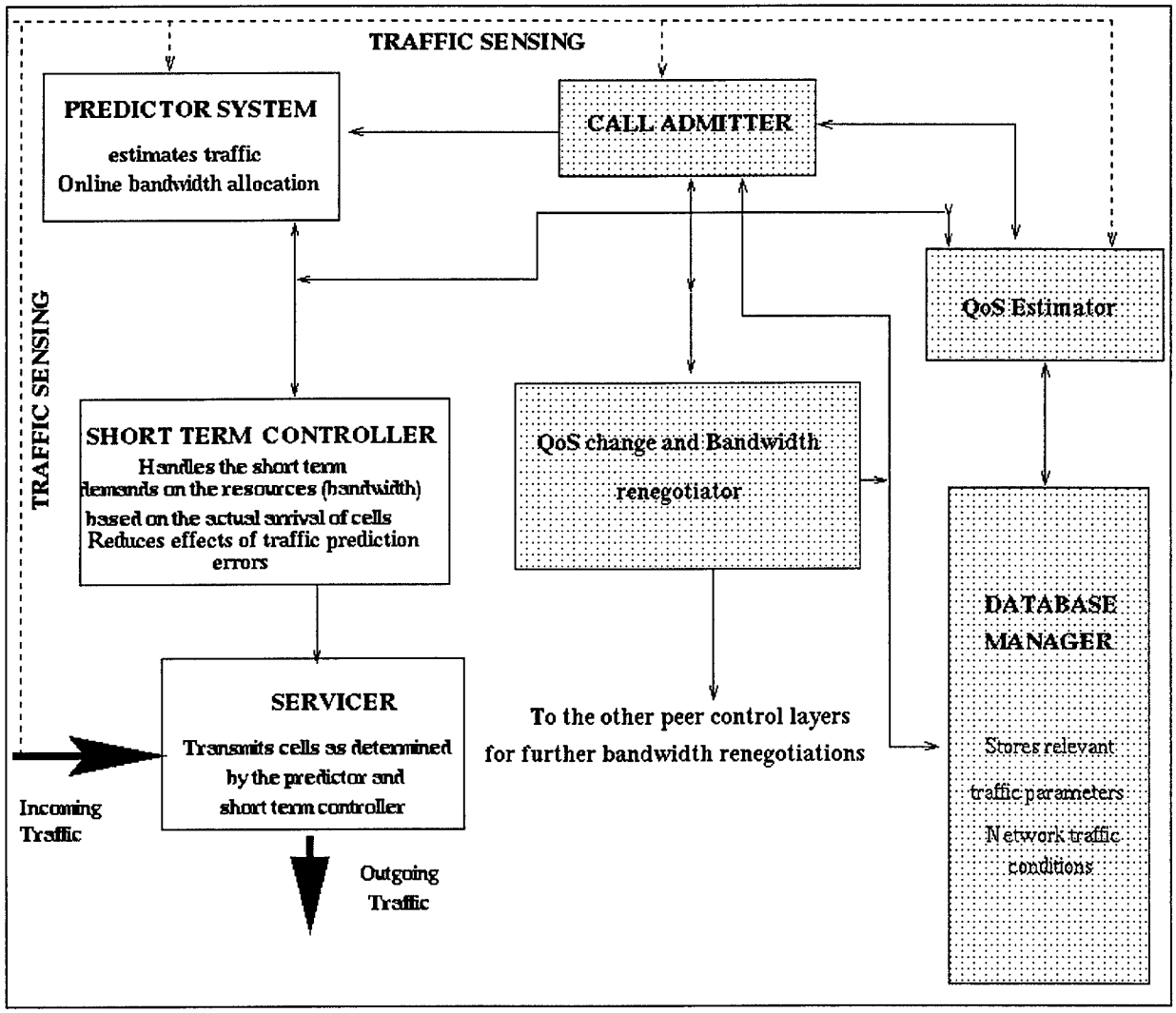


Figure 1. Integrated Framework for Resource Management.

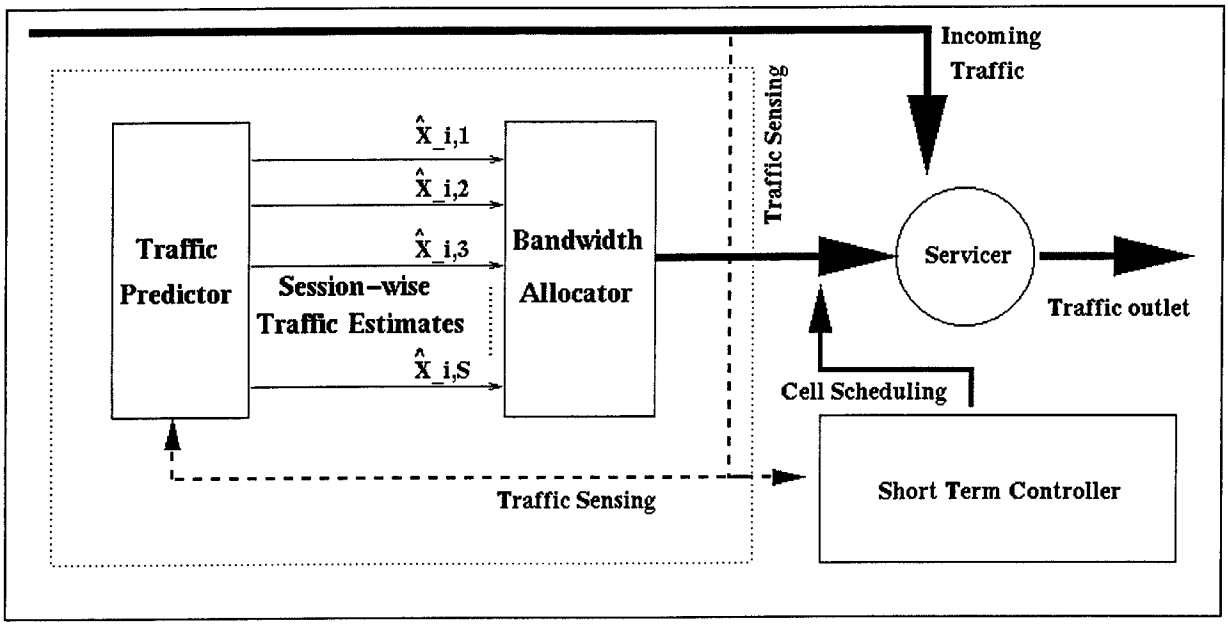


Figure 2. The Predictor-STC System.

10. Conclusion

An integrated and intelligent network management platform for efficiently allocating resources and optimally adapting to the communication networks environment was developed and it is presented in this report. The concept of performance trending technique is introduced. The investigation shows that *soft* network failures can be predicted by real-time monitoring, analyis, and performance trending of QoS parameters. The network recovery before any loss of communication by hand-over to an alternate network using a fuzzy rule-based system is demonstrated. Further, the issues in resource management of video traffic were examined. A predictor-based architecture with a short-term controller for dynamic bandwidth management in an ATM environment is described.

Acknowledgements

The author wishes to express his sincere appreciation to the Air Force Office of Scientific Research (AFOSR) for sponsoring the summer research program. Special thanks to the focal point, Mr. Robert Kaminski, Communications Networks Branch, Rome Laboratory for sponsoring and providing the facility to conduct this research project.

References

[1] D. McDysan and D. Spohn, *ATM: Theory and Application*, McGraw-Hill, 1995.

[2] Z. Dziong, *ATM Network Resource Management*, McGraw-Hill, 1997.

[3] K. Terplan, *Communication Networks Management*, Prentice-Hall, 1987.

[4] A. Leinwand and K. F. Conroy, *Network Management: A Practical Perspective*, Addison-Wesley, 1996

[5] R. Sankar and R. Singh, "Performance Trending for Management of Packet-switched Networks," *International Journal of Network Management*, Vol. 4, pp. 69-78, Jun. 1994.

[6] G. Chiruvolu, K. J. Christensen and R. Sankar, "Short-term Resource Management for Better ATM QoS," *Sixth Int. Workshop on Network Operating System Support for Digital Audio and Video*, pp. 31-34, Apr.1996.

[7] G. Edwards and R. Sankar, "Handoff using Fuzzy Logic," *Proc. IEEE Globecom*, pp. 524-528, Nov. 1995.

[8] G. Edwards and R. Sankar, "Fuzzy Control for Microcellular Corner Effect," *Proc. International Conf. on Fuzzy Systems*, pp. 1912-1916, Sep. 1996.

[9] L. Zadeh, "Fuzzy Logic," *IEEE Computer*, pp. 83-93, Apr.1988.

[10] L. Wang, *Adaptive Fuzzy Systems and Control*, Prentice-Hall, 1994.

[11] G. Chiruvolu and R. Sankar, "An Approach Towards Resource Management and Transportation of VBR Video Traffic, *Proc. IEEE ICC*, pp. 550-554, Jun. 1997.

[12] S. Chong, San-qi Li, and J. Ghosh, "Predictive Dynamic Bandwidth Allocation for Efficient Transport of Real-time VBR Video over ATM," *IEEE JSAC*, Vol. 13, pp. 12-23, Jan. 1995.

ERRORS INHERENT IN RECONSTRUCTION OF TARGETS FROM MULTI-LOOK IMAGERY

Mark S. Schmalz
Associate Professor
Department of Computer Information Science and Engineering

University of Florida
Bldg. CSE/E301
Gainesville, FL 32611

Final Report for:
Summer Research Program
Rome Laboratory

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, Washington, DC

And

Rome Laboratory

September 1997

ERRORS INHERENT IN RECONSTRUCTION OF TARGETS FROM MULTI-LOOK IMAGERY

Mark S. Schmalz
Assistant Professor
Department of Computer and Information Science and Engineering
University of Florida

Abstract

Automated target recognition has benefited from cross-fertilization of development in related subdisciplines of image processing such as medical imaging. For example, the application of computerized tomography to synthetic aperture radar (SAR) imaging has produced 3-D reconstructions of ground targets on an experimental basis. In practice, by acquiring multiple views of a target (also called *multi-look imaging -- MLI*) that are subsequently merged mathematically, one can obtain reasonable approximations to higher-dimensional reconstructions of a target of interest. For example, multiple two-dimensional airborne images of ground objects can be merged via the Fourier transform (FT) to obtain one or more approximate three-dimensional object reconstructions. Additional methods of 3D model construction (e.g., from affine structure) present advantages of computational efficiency, but are sensitive to positioning errors.

In this study, an analysis of MLI is presented that applies to various scenarios of nadir, near-nadir, or off-nadir viewing with a small or large number of narrow- or wide-angle views. A model of imaging through cover describes the visibility of a given target under various viewing conditions. The model can be perturbed to obtain theoretical and simulated predictions of target reconstruction error due to (a) geometric projection error, (b) focal-plane quantization error and camera noise, (c) possible sensor platform errors, and (d) coverage of looks. An information-theoretic model is derived from the imaging model that can facilitate prediction of limiting sensor geometry and view redundancy under various imaging constraints (e.g., target and cover geometry, available range of look angles, etc.). Ongoing research emphasizes selection or design of target reconstruction algorithms based on estimated MLI data and systematic errors. Additional discussion concerns the use of physical models for facilitating inexpensive (< $150) acquisition of image data for preliminary MLI algorithm testing.

Study notation is a subset of image algebra, a rigorous, concise, computationally complete notation that unifies linear and nonlinear mathematics in the image domain [Rit96]. Image algebra was developed at University of Florida over the past decade under the sponsorship of DARPA and the US Air Force, and has been implemented on numerous sequential workstations and parallel processors. Hence, our algorithms are rigorous and widely portable.

ERRORS INHERENT IN RECONSTRUCTION OF TARGETS FROM MULTI-LOOK IMAGERY

Mark S. Schmalz

1. Introduction

Automated recognition of ground targets from airborne imagery can be confounded by vegetative or manmade cover that overlies targets of interest. For example, consider a vehicle parked beneath an overhanging tree line or concealed under camouflage netting. Under such circumstances, it would be advantageous to acquire and exploit multiple target images at different look angles (called *multi-look imaging* or *MLI*) [Lon81,Chr96,Moo96,Sha96,PAR97]. Selected MLI images could be combined mathematically to yield a composite view of a target of interest. With appropriate supporting models, this process is being implemented in computer software that produces simulated MLI imagery and error analyses [Sch97a,b].

In this study, the following objectives are accomplished:

1) Understand and model the phenomenology of multi-look imaging;
2) Develop models and error analyses for tomographic or tomography-like target reconstruction from multiple airborne views of ground targets;
3) Apply theory and models to the development of computer software that estimates errors in salient system parameters under given sensing geometry constraints; and
4) Acquire realistic laboratory or field imagery to facilitate algorithm design and test.

Computer model(s) are summarized in Section 4.

1.1. Document Organization

This report begins with a review of previous work (Section 1.2) and a summary of key study issues and assumptions (Section 2), followed by a presentation of theory and methodology that describes basic MLI phenomenology (Section 3). Theory is extended to yield an information-theoretic model of target visibility in MLI scenarios, which supports theory development for tomographic reconstruction. The result of this study, a preliminary analysis of the difficult and error-prone process of target reconstruction from multiple views is discussed in Section 4. Conclusions and suggestions for future work are presented in Section 5.

1.2. Previous Work

Detection and recognition of manufactured targets in natural scenes is based on the selection of a model that best matches a subregion of an observed image. When an imaging sensor's view of a target is unobstructed, the matching process can be implemented via 1) least-squares minimization in image space, or 2) suboptimal methods using correspondence subsets. In the presence of partial occlusion, one could 3) progressively reconstruct the object via coregistration and merging of multiple views or 4) employ tomography-like methods that approximate a solid model of the target. The tomographic model could be serially sectioned with respect to a reference vector, thereby yielding a layered model of objects located between ground and cover. Methods 1) through 3) are summarized in Sections 1.2.1-1.2.3, followed by comparison of three-dimensional (3-D) models and 2-D images in Section 1.2.4. We do not consider the limiting and extremely difficult case of total occlusion, which is outside the scope of this study.

1.2.1. Least-squares minimization techniques. Traditional photometric approaches to matching images and reference models involve least-squares minimization of the Euclidean distance between the model and an elevation map con-

structed from the image. This problem is generally classified as an *exterior orientation calibration problem* (EOCP) whose solution, given a set of 3-D (2-D) model (image) points **U** (**V**), finds a rigid transformation that minimizes a distance between **U** and **V** [Bas96]. Since analytical solutions to the EOCP remain undiscovered, numerical methods are employed, which tend to be unstable in the presence of discontinuities and noise, are computationally costly, and require accurate initial values.

Marr and Nishihara [Mar78] first addressed the numerical problems of describing 3-D structure systematically. Minsky [Min75] had previously suggested that the complexity of shape description could be minimized by appropriate choice of primitives, although this has yet to be realized in general theory. In the intervening two decades, surface reconstruction from a single view has been implemented in terms of variational methods [Mei79,Sch77,Ter83,Ter88], segmentation and synthesis or a variety of ad hoc techniques [Bes85]. Bolle and Vemuri [Bol91] summarize variational surface fitting techniques that (a) have varying degrees of viewpoint invariance as well as robustness in the presence of noise, (b) tolerate bias in parameter estimates, and (c) exhibit various sensitivities to obscuration. The following discussion is derived from the excellent overviews given in [Bol91] and [Vem86,87].

In its simplest form [Bar81], surface fitting involves interpolation of uniformly curved surfaces from initial orientation values and constraints using a relaxation algorithm based on parallel iterative local averaging. Unfortunately, this technique ignores surface discontinuities, is computationally costly, and produces a surface representation that is invariant to 3-D rigid motion.

Grimson [Gri81] presented a theory of visual surface interpolation based on range data from stereo imagery that were obtained via Marr and Poggio's correspondence algorithm [Mar79]. Given a finite image domain $\mathbf{X} \subset \mathbf{R}^n$ and an elevation map $\mathbf{a} \in \mathbf{R}^{\mathbf{X}} \equiv \mathbf{X} \rightarrow \mathbf{R}$, Grimson minimized the variation of the quadratic functional

$$E = \int_N \mathbf{a}_{xx}^2 + 2\mathbf{a}_{xy}^2 + \mathbf{a}_{yy}^2 \; dx \; dy \qquad \text{(I)}$$

where $(x, y) \in \mathbf{X}$, $N \subset \mathbf{X}$, and $\mathbf{a}_{xy} \equiv \partial x / \partial y$. E denotes the energy of a thin plate, hence its minimization yields a function called *thin plate splines* [Mei79]. Grimson developed an iterative algorithm based on biharmonic convergence that results from applying Euler's equations to minimize E. Although invariant to rotation and translation, Equation (I) is not invariant to viewpoint, since E depends on the coordinate system in which depth constraints are specified. Additionally, Grimson's model did not admit discontinuities.

In contrast, Terzopoulos [Ter83,88] derived an efficient surface model that accepts multiresolution imagery, is computationally efficient, and accounts for depth and orientation discontinuities. Surfaces are modeled as thin plate segments joined by membrane strips along orientation discontinuities and bounded by depth discontinuities. The model is invariant to translation and rotation but not to rigid motion, and is based upon a continuity stabilizer described in [Ter83]. Harris' extension [Har86] of Terzopoulos' model couples depth and slope, this integrating orientation constraints into the minimization of the energy functional E. Harris' model admits varying surface smoothness, parallel implementation, discontinuities of any order, and generalizes splines under tension since it can integrate arbitrary combinations of membrane, thin-plate, or high-order smoothness constraints.

In MLI imagery, the surface fitting method would be complicated by the lack of registration between views that could exhibit geometric distortion, rotation, and scale differences. Hence, a parameterized method (e.g., quadric or superquadric representation) would be required that could be scaled, rotated, and translated to implement coregistration between views in model space (versus customary image-domain coregistration). Sensor noise, projection error,

and quantization error would thus increase surface fitting error. More importantly, the greyscale variations induced by patterns of light and shadow or camouflage devices could easily confound the construction of an elevation map based on well-known shape-from-shading techniques, as discussed in [Sch97b].

1.2.2. Suboptimal matching with correspondence subsets. The change in the 2-D projection of a moving 3-D object or camera yields important information for 3-D object reconstruction that elucidates relative geometry. The problem of computing shape from motion or a sequence of images has been addressed using calibrated or uncalibrated cameras [Fau92] as well as projective or affine reconstruction models [Fau95]. Camera calibration facilitates the computation of Euclidean shape up to a scale factor using projective or affine models [Der94]. In the absence of calibration, the recovered shape is defined up to a projective or affine transformation. Although weaker than that provided by surface fitting from depth or elevation maps, or by inversion of projection equations with fully specified camera calibration matrices, relative geometry is useful since

1. Reconstruction algorithms can be made simpler and more easily parallelized;
2. The costly and time-consuming process of camera calibration is not required;
3. Multiple views of an object need not be equally spaced to obtain regularity; and
4. Distinction between orthographic and perspective projections is not required.

In practice, one (a) chooses a representation of projective space in which an arbitrary reference plane is assumed to be located at infinity, (b) describes the new representation by an element of an affine group applied to the initial representation, and thus (c) obtains an affine invariant relative to the initial representation.

Affine reconstruction is expected to be a useful target reconstruction technique in conjunction with MLI. Three recent papers highlight associated technical issues:

- *Christy & Horaud* [Chr96] incrementally perform Euclidean reconstruction based on a weak-perspective or paraperspective (calibrated) camera model. Fast convergence (in a few iterations), computational efficiency, and solution of the sign reversal ambiguity problem are cited as advantages.
- *Moons et al.* [Moo96] describe affine reconstruction of a scene from two views with relative target-camera translation between views. The reported approach requires only five corresponding points, which can be obtained by well-known coregistration algorithms [Yes89,Smi96], unless significant anisomorphic distortion is present in coregistered views. This is usually the case for multi-look imagery where a majority of the views are taken over a narrow angular range.
- *Shashua & Navab* [Sha96] describe an affine reconstruction technique for perspective views that unifies Euclidean, projective, and affine models. Simple algorithms are presented for reconstruction from multiple views, recognition by alignment of segmented objects with templates, and coding of video sequences.

We see affine reconstruction as a possible alternative to tomographic reconstruction that is attractive implementationally due to viewpoint invariance [Sha96], lack of camera calibration requirements [Sha96, Moo96], and tolerance of camera translation [Moo96]. More detailed discussion is presented in [Sch97b].

Shashua and Navab analyzed interpolation and extrapolation errors. Given three views $\mathbf{a}_{i-1}$, $\mathbf{a}_i$, and $\mathbf{a}_{i+1}$, extrapolation based on the (i-1)-th and i-th views was employed in reconstructing an approximation to the (i+1)-th view. Error was measured as the mean squared distance between corresponding points in the extrapolated and actual views. As expected, the upper bound on error was provided by extrapolation, whose error decreased to 1.1 ± 0.98 pixels as the separation between the i-th and (I+1)-th views increased, up to the visibility limit. This is due to the fact that ex-

trapolation is usually more erroneous than interpolation, given identical continuity assumptions. Scene reconstruction error was measured as the mean-squared difference between predicted and measured object elevation. A cube having greyscale patterns on its faces was employed as a test object. The average depth error was 0.23 ± 0.31 percent of mean range distance r to target, with a peak-to-peak error range of 0.002r to 0.007r.

Practical considerations in affine-based 3-D modeling for MLI applications emphasize the accuracy with which the various points can be located in terms of world or relative coordinates. In imagery, pointing accuracy is limited by focal plane quantization error. The accurate determination of sensor position and orientation is crucial to establishing the camera origin points. If the separation between sensor origins can be determined accurately, then Shashua's method [Sha96] can be modified to yield displacement-based measures of relative affine structure, per [Moo96].

1.2.3. Coregistration-based techniques. Another approach to multi-look imaging involves coregistration of the multiple looks across projections (views) and spectral bands (in the case of multispectral imagery). If the spectral bands are not severely decorrelated temporally, current coregistration technology may be adequate for inter-band alignment within ± 1 pixel. However, each projection (taken at a different look angle and altitude) has scaling, rotation, shift, and anisomorphic geometric distortions that tend to make registration across all views difficult for wide-angle MLI.

Yeshurun and Schwartz first proposed cepstral filtering for region-based solution of the stereo correspondence problem [Yes89], which instantiates a correspondence problem. This scheme uses an adaptive prefilter followed by autocorrelation, which provides prewhitening that reduces the noise sensitivity of the resulting stereo disparity. Given a stereo pair of MxN-pixel images, the stereo image formed as $\mathbf{c} = (\mathbf{a},\mathbf{b})$ is transformed via application of the cepstrum to yield the disparity image $\mathbf{d}$, as follows:

$$\mathbf{d} = \mathcal{F}(\log[|\mathcal{F}(\mathbf{a}) + 1|]) \quad ,$$

where $\mathcal{F}$ denotes the Fourier transformation. The difference between the coordinates of the origin and largest peak of $\mathbf{d}$ yields the stereo disparity. Since the cepstral approach can be applied to small neighborhoods of an image that contain corresponding points, it is useful in principle for imagery that has space-variant distortion. Unfortunately, the location of correspondence neighborhoods is itself an instance of the correspondence problem that may simplify the problem implementationally via a hierarchical registration scheme, but does not fundamentally reduce complexity.

Ludwig et al. [Lud94] performed an error analysis of Yeshurun and Schwartz' technique, to determine the performance of cepstral-based correspondence as a function of image noise and distortion. It was suggested that LoG windowing functions be employed for vignetting each region to which the cepstrum is applied. This reduces disparity artifacts slightly without improving algorithm robustness in the presence of scaling distortion or photometric variance.

Smith and Nandhakumar [Smi96] employ the cepstrum as a nonlinear correlator to achieve increased tolerance of noise and anisomorphic geometric distortions. Their enhancements to Yeshurun and Schwartz' technique (a) replace the peak detection process in the power cepstrum with a more robust search mechanism in disparity space and (b) correct for foreshortening effects. The result is a reduction in sensitivity of the corresponding disparity measure to photometric variation and anisomorphic distortions.

Given the foregoing techniques, advantages of coregistering MLI images include:

- Effects of random sensor noise can be ameliorated somewhat by accumulation of the coregistered images, and
- One may be able to obtain depth information for common points across all views of an MLI image sequence, for purposes of 3-D model construction.

Unfortunately, the following practical disadvantages can accrue from erroneous statistical assumptions, computational error, and continuity assumptions required for coregistering multiple views:

- Lateral spatial alignment (coregistration) error may induce elevation errors in stereo reconstruction of depth maps from MLI sequence pairs, which errors may exceed salient target feature height; and
- Computational error inherent in the resampling process required to rectify (coregister) imagery to compensate disparity effects may also induce elevation errors in the reconstructed 3-D target model.

Although region-based cepstral correspondence matching is currently our method of choice for image coregistration, additional research remains in global coregistration techniques. Further discussion and error analyses are given in [Smi96] and [Yes89].

1.2.4. Comparison of 3-D models and 2-D images. Given an image **a**, an object recognition process typically extracts key features, then seeks a model M that best matches the closest view of the feature ensemble [Fis81,Hut90,Ull91]. Image-model alignment can be inexact due (for example) to input noise, model specification errors, and deformation in nonrigid objects. Thus, robust assessment of alignment accuracy is required. The common assumption of a Euclidean distance (called the *image metric* [Bas96]) between image feature points and their corresponding points in the model view of an object that best matches imagery assumes that images (being inherently more erroneous than models) induce alignment perturbations that should be measured in the image plane. Although this assumption may be suitable for recognition, object classification tends to benefit more from minimal object deformation that tolerates uncertainties in a test object's structure.

Measures that compare 3-D models and 2-D images should be *metrical*, which Basri [Bas96] defines as increasing monotonically with the difference between M and **a**. To achieve this, the class of transformations applied to the object are customarily extended from rigid to affine transforms. Although this bounds the rigid measure from below, an upper bound is not attained. Previous methods tended to achieve suboptimal distances or did not produce a metric.

Basri and Weinshall [Bas96] reported a distance measure f that compares a 3-D model M with a 2-D image **a** in closed form. This penalizes nonrigidity incurred by an optimal affine transformation T that aligns M with **a** under weak-perspective projection, as follows:

$$f(M, \mathbf{a}) = \bigwedge_{R \in \mathrm{T}_{R_g}} \| T(M, \mathbf{a}) - R(M, \mathbf{a}) \|^2 ,$$

where T_{Rg} denotes the set of all rigid transformations and the norm is denoted by (|| ||). Thus f, which bounds the least-squares distance from above and below, facilitates

1. Direct assessment of similarity between 3-D models and 2-D images;
2. Obtaining upper and lower bounds on the *image metric* for object classification; and
3. Derivation of an initial value set for numerical surface fitting procedures.

An additional application is the evaluation of hypothesized correspondences in *alignment algorithms*, which evaluate the similarity between models and images based on sparse point-to-point correspondences. Although this technique can produce polynomial-time matching in the presence of occlusion [Gri92], matching errors invariably result due to sparse sampling, which can be mitigated with additional correspondences [Fis81]. The measure f can be further employed to evaluate the amount of affine distortion applied to an object relative to a prototype, thereby determining the best-match object class or, in certain cases, object identity.

2. Key Issues and Assumptions

We assume that there exist the following three primary scenarios for MLI:

1. *Narrow-angle, Few Looks* -- the sensor look angle is perturbed only slightly from a reference orientation

(customarily nadir viewing) and a few (e.g., 5 to 10) images are taken over a relatively narrow field-of-view (e.g., 10 to 20 degrees).

2. *Wide-angle, On-track* -- the sensor moves along an approximately linear or curvilinear track, and acquires imagery over few or many looks, over a wide field of view.
3. *Loitering* -- the target is reconnoitered by an airborne sensor describing a simple or complex curvilinear path, with unequally spaced views from different elevations.

Scenario 1 has the potential of producing approximately equally spaced looks, some of which can be corrupted by cover interposed between sensor and target. This is the scheme proposed in [PAR97], upon which tomographic reconstruction of the target would be based. Scenarios 2 and 3 would typically acquire unevenly spaced images, with scale differences likely as sensor altitude (and, possibly, camera magnification) changes unpredictably. We have found that such schemes are often amenable to 3D target reconstruction via affine structure extraction or surface fitting. Tomography-based approaches would be confounded by large partitions of missing information in the tomographic reconstruction space, especially if few looks were taken.

3. Methodology

We begin with a one-dimensional model, for purposes of illustration, then progress to a two-dimensional model of cover, which is instantiated in computer software.

3.1. Theory of Multi-Look Imaging

Assumption 1. Let a spatial domain $\mathbf{X} \subset \mathbf{R}$ represent (for example) a background region viewed by an imaging sensor at altitude a_s, as shown schematically in Figure 1. Let opaque cover located at altitude a_c be comprised of segments having width w_i, where $1 \le i \le n$. The segments are separated by horizontal apertures g_i that have centroids at $x_i^c \in \mathbf{X}$. Per Figure 1, we assume that the i-th aperture precedes the i-th cover segment as one proceeds away from the sensor.

Lemma 1. *Given Assumption 1, if the sensor images a contiguous partition* $[x_i^{min}, x_i^{max}]$ *of* $\mathbf{X}$ *through aperture , then the following statements hold:*

1. *Look angle extrema* $\phi_v^{min}(i) = \tan^{-1}\left(\dfrac{\sum_{j=1}^{i-1} g_j + w_j}{a_s - a_c}\right)$ *and* $\phi_v^{max}(i) = \tan^{-1}\left(\dfrac{g_i + \sum_{j=1}^{i-1} g_j + w_j}{a_s - a_c}\right)$;
2. *Partition extrema* $x_i^{min} = a_s \cdot \tan(\phi_v^{min})$ *and* $x_i^{max} = a_s \cdot \tan(\phi_v^{max})$; *and*
3. *Field of view* $v_i = x_i^{max} - x_i^{min}$.

Proof. The proof follows directly from the geometry of Figure 1. ■

The preceding model does not suffer loss of generality, since if all indices i are in the interval $[1,n]$, then $w_i = 0$ depicts the case of no cover and describes the situation where there are no apertures. Alternatively, if a given aperture subtends $\mathbf{X}$, then there is no cover but viewing is unobstructed. In practice, the model of Figure 1 is best applied when a target is situated against the background and the cover segments w_i are vertically thin in relation to the sensor or cover altitude. We also assume that the apertures g_i are sufficiently large to render diffraction effects negligible.

Additionally, the preceding one-dimensional model can be generalized to higher-dimensional domains by observing that, in multi-look imaging, the received image **a** is comprised of more than one projection of the target restricted to the corresponding projection of the cover. Since the cover is assumed to be opaque, application of restriction in this context does not compromise rigor. Hence, the one-dimensional model and its generalization in two or more dimensions are sufficiently general and rigorous to portray physical reality in this preliminary study.

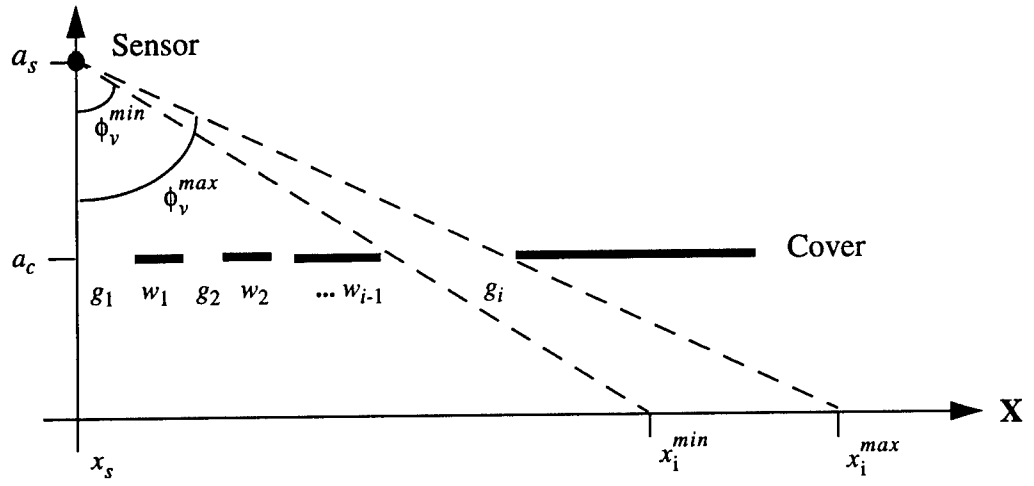


Figure 1. Schematic sensor geometry for multi-look imaging through perforated cover, where x_s denotes lateral sensor position.

Assumption 2. Let domain $\mathbf{X} \subset \mathbf{R}^2$ and assume that cover is specified by an image $\mathbf{c} \in \mathbf{R}^{\mathbf{X}}$ located a_c meters (m) above a target whose elevation map (or equivalent projection) is denoted by the Boolean image $\mathbf{b} \in \mathbf{B}^{\mathbf{X}}$. In this study, we adopt the convention that the values of **c** denote the transmission coefficients of the cover. Let an imaging sensor of focal length f and magnification M be located a_s meters above the target plane, as shown in Figure 2. For each configuration of cover $\mathbf{c}_i$ and target $\mathbf{b}_i$, where $1 \le i \le n$, let the sensor form an image $\mathbf{a}_i \in \mathbf{F}^{\mathbf{X}}$. Further assume that the transmissive medium between target and cover (cover and sensor) has absorption and scattering coefficients a_{tc}, b_{tc} (a_{cs}, b_{cs}) and volume scattering function β_{tc} (β_{cs}).

Observation. When simulating the formation of sensor image **a**, one of two cases pertain:

Case 1: A flat target (or one that can be assumed to be flat), where **b** denotes a reflectance map, or

Case 2: 3-D target, where **b** denotes an elevation map.

For purposes of illustration, the following development is specific to the first case, and the second case is discussed beginning with Theorem 3.

Definition 1. A *spatial transformation* $f : \mathbf{Y} \to \mathbf{X}$ is a map between spatial domains. Given an image $\mathbf{a} \in \mathbf{F}^{\mathbf{X}}$, a spatially transformed image $\mathbf{b} \in \mathbf{F}^{\mathbf{Y}}$ is denoted by

$$\mathbf{b} = \mathbf{a} \circ f \equiv \{(\mathbf{y}, \mathbf{a}(\mathbf{y})) : \mathbf{y} \in \mathbf{Y}\} .$$

Theorem 1. *Given Assumption 1 and the geometry of Figure 2, the projection S of a point* $\mathbf{x} \in domain(\mathbf{c})$ *to a point* $\mathbf{y} \in domain(\mathbf{d})$ *is given by*

$$\mathbf{x} = S(\mathbf{y}) = \left(\frac{a_s - a_c}{a_s}\right) \cdot h' \cdot (\sin\theta_v, \cos\theta_v) ,$$

where $h' = ([p_1(\mathbf{y})]^2 + [p_2(\mathbf{y})]^2)^{1/2}$ *and* $\theta_v = \tan^{-1}(p_2(\mathbf{y})/p_1(\mathbf{y}))$, *with* $p_k(\mathbf{y})$ *denoting projection to the k-th coordinate, which implies that* $\mathbf{d} = \mathbf{c} \circ S$.

Proof. From the similar triangles of Figures 1 and 2, the target dimension along the *x*- and *y* -axis of *domain*(**b**) is scaled by $(a_s - a_c)/a_s$ in *domain*(**c**). This scale factor is also applied to the hypotenuse h' to yield h in *domain*(**c**). The hypotenuse h' is projected via angle θ_v which does not vary between **b** and **c**. Hence, we have the expression $S(\mathbf{y}) = ((a_s - a_c)/a_s) \cdot h' \cdot (\sin\theta_v, \cos\theta_v)$, and $\mathbf{d} = \mathbf{c} \circ S$ follows from Definition 1. ■

Theorem 2. *Given a volume scattering function* $\beta : (-\pi, \pi] \rightarrow [0, 1]$ *and a scattering distance* $d \in \mathbf{R}^+$ *then a template* $\mathbf{s} : \mathbf{R}^+ \times \mathbf{X} \rightarrow \mathbf{R}^{\mathbf{X}}$ *that instantiates a single-scattering, near-field approximation of* $\beta(\vartheta)$ *is defined in terms of its weights as*

$$\mathbf{s}_{\mathbf{y}}(d)(\mathbf{x}) = \beta(\vartheta), \text{ where } \vartheta = \tan^{-1}\left(\frac{\|\mathbf{y} - \mathbf{x}\|}{d}\right) .$$

Proof. Given the geometry of Figure 3, it suffices to state that $\vartheta = \tan^{-1}(\|\mathbf{y} - \mathbf{x}\|/d)$. ■

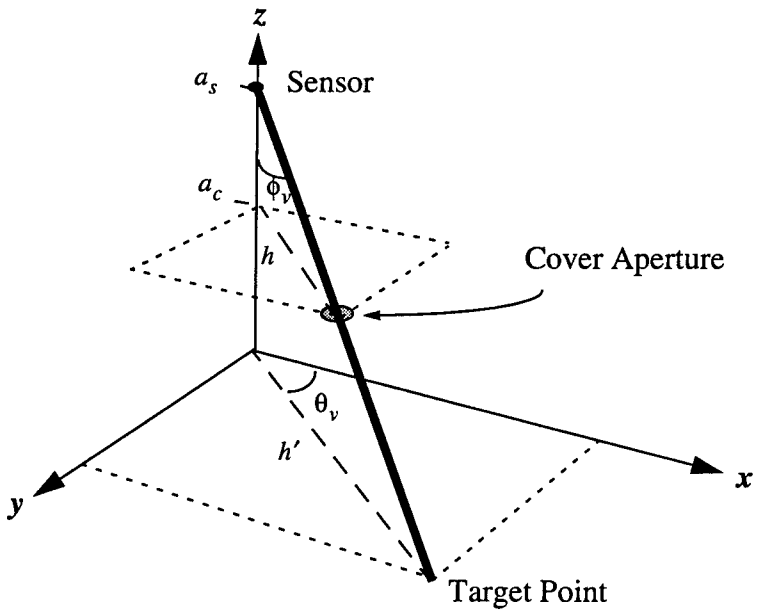


Figure 2. Three-dimensional sensing geometry.

3.2. MLI Model

We next present several algorithms and theorems that we are currently instantiating in a computer model, which would simulate the effect of multiple views of a target object through perforated cover.

Algorithm 1. Given Assumption 1, if **b** denotes a greyscale image (e.g., a reflectance map), then the formation of image $\mathbf{a}_i$ can be simulated by performing the following steps:

Step 1. Form the spatial transform $S_{ii} : \mathbf{X} \rightarrow \mathbf{X}$, from which the projection of $\mathbf{c}_i$ to the target plane is given by $\mathbf{c} \circ S$, per Definition 1. From *S*, sensor geometry, and the geometry of **X**, one obtains a map $\mathbf{d} \in (\mathbf{R}^+)^{\mathbf{X}}$

of the propagation distance from the target plane to the sensor as a function of a point $\mathbf{x} \in \mathbf{X}$, as shown in Theorem 2.

Step 2. Given the media optical parameters a_{tc}, b_{tc}, and β_{tc}, form the attenuation template $\mathbf{s}_{tc} : \mathbf{R}^+ \times \mathbf{X} \rightarrow \mathbf{R}^{\mathbf{X}}$, which is parameterized by the propagation distance between target and cover, per Theorem 2. An additional attenuation template $\mathbf{s}_{cs}$ is formed symmetrically from a_{cs}, b_{cs}, and β_{cs} that portrays attenuation in the propagation path segment from cover to sensor.

Step 3. Apply S, $\mathbf{s}_{tc}$, and $\mathbf{s}_{cs}$ to $\mathbf{c}$ to yield $\mathbf{s}_i = \mathbf{s}_{tc}(\mathbf{d}_i) \oplus \mathbf{s}_{cs}(\mathbf{d}_i)$ and an approximation to $\mathbf{a}_i$, as follows:

$$\mathbf{a}_i = [\mathbf{b} * (\mathbf{c}_i \circ S_i)] \oplus \mathbf{s}_i , \qquad \text{(II)}$$

The preceding equation does not include realistic receiver effects, such as noise, nonuniform pixel gain, or missing pixels, which we have discussed in [Sch96a], and which would be included in the completed model.

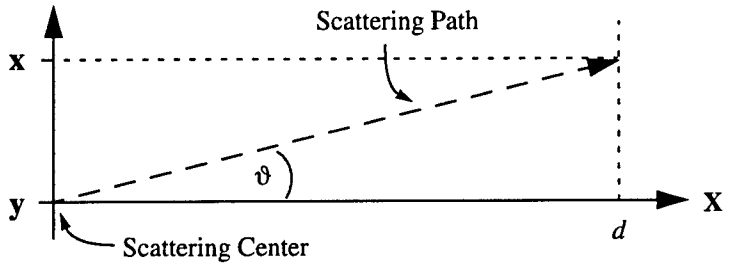


Figure 3. Linear approximation for single-scattering geometry.

Theorem 3. *Let a cylindrical target of radius r and height h be viewed by an ideal camera of unitary magnification, at look angle ϕ measured away from the vertical). The apparent target dimension is given by $h \cdot \cos\phi + r \cdot \sin\phi$.*

Proof. The proof follows directly from the construction of Figure 4a. ■

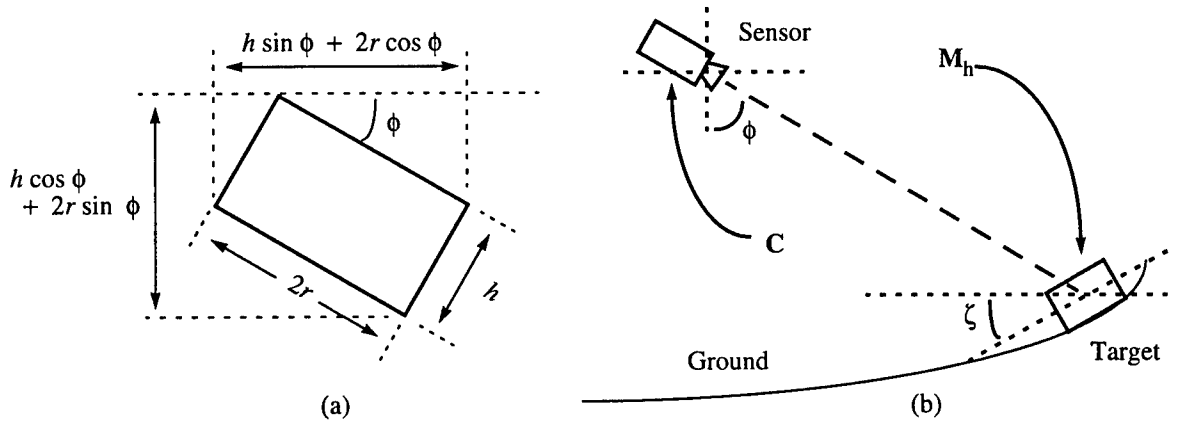


Figure 4. Three-dimensional target (a) geometry, and (b) sensing configuration.

Theorem 4. *Given the geometry of Figure 4b, if* $\mathbf{P}$ *denotes a perspective correction matrix;* $\mathbf{C}$, $\mathbf{R}$, *and* $\mathbf{G}$ *denote camera, rotation, and translation matrices, respectively; and* $\mathbf{M}_h$ *denotes local target coordinates; then the target coordinates projected to the focal plane of a sensor at altitude a_s are given by* $\mathbf{C} = (x, y, z, 1)^T$, *such that* $\mathbf{C}_f = \mathbf{P}\,\mathbf{C}\,\mathbf{R}\,\mathbf{G}\,\mathbf{M}_h$.

Proof. Let $\mathbf{M}_h$ denote local target coordinates, and let translation matrices **G** and **C** be given by

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & 0 & -p_1(\mathbf{p}) \\ 0 & 1 & 0 & -p_2(\mathbf{p}) \\ 0 & 0 & 1 & -p_3(\mathbf{p}) \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{C} = \begin{pmatrix} 1 & 0 & 0 & -p_1(\mathbf{p}) \\ 0 & 1 & 0 & -p_2(\mathbf{p}) \\ 0 & 0 & 1 & -p_3(\mathbf{p}) \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

where **r** denotes the coordinates of the sensor camera's focal point, with the camera lens nodal point located at **p**. Let $\mathbf{R} = \mathbf{R}_\phi \mathbf{R}_\theta$ be a combination of two rotations $\mathbf{R}_\phi$ (tilt angle) and $\mathbf{R}_\theta$ (pan angle), such that

$$\mathbf{R} = \begin{pmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta\cos\phi & \cos\theta\cos\phi & \sin\phi & 0 \\ \sin\theta\sin\phi & -\cos\theta\sin\phi & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Define the perspective projection matrix **P** as follows:

$$\mathbf{P} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1/f & 1 \end{pmatrix},$$

where f denotes the sensor focal length. Assume that translation and rotation matrices **T** and **R** are given by

$$\mathbf{T} = \begin{pmatrix} 1 & 0 & 0 & -x_m \\ 0 & 1 & 0 & -y_m \\ 0 & 0 & 1 & -z_m \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{T}^{-1} = \begin{pmatrix} 1 & 0 & 0 & x_m \\ 0 & 1 & 0 & y_m \\ 0 & 0 & 1 & z_m \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \text{and} \quad \mathbf{R} = \begin{pmatrix} 1 & 0 & 0 & -x_m \\ 0 & \cos\zeta & \sin\zeta & -y_m \\ 0 & -\sin\zeta & \cos\zeta & -z_m \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

From the expression of Theorem 3 in two dimensions, and letting r and h denote the radius of height of a cylindrical target inclined at an angle ζ from the horizontal, we have that $\mathbf{M}_h = (h\sin\phi + 2r\cos\phi,\ h\cos\phi + 2r\sin\phi,\ h_0)$, where $\phi \in [0, 2\pi)$ and $|h_0| \le h/2$. One thus obtains

$$\mathbf{W}_h = \mathbf{T}^{-1}\mathbf{R}\,\mathbf{T}\,\mathbf{M}_h,$$

which yields the projection of target coordinates to image space as

$$\mathbf{C} = \mathbf{P}\,\mathbf{C}\,\mathbf{R}\,\mathbf{G}\,\mathbf{M}_h. \qquad \blacksquare$$

3.3. Theory of Tomographic Reconstruction

Computerized axial tomography customarily projects a planar beam of X-rays through an object, where the transmitted X-rays are measured by a linear array of detectors. By rotating the transmitter/detector through 180 degrees at small, fixed angular increments then applying a reconstruction algorithm, a high-resolution of a given cross-section through an object may be obtained. Combining these cross-sections into a solid model yields 3-D information concerning the object of regard. In this study, we concentrate on airborne MLI imagery of battlefield scenes, and investigate whether or not tomographic imaging techniques could be applied to such imagery. We call this process

battlefield tomography, and note that control of the sensing parameters or knowledge of target/media properties would not necessarily be precise. In contrast, medical tomography has the advantages of (a) well-known optical parameters of various tissues that vary with relative predictability cross-population and among various pathological conditions, (b) a highly constrained geometry where transmitter and detector positions are known accurately, (c) quantization bin-size that can be set sufficiently small to meet diagnostic accuracy constraints, and (d) offline processing capability that admits reasonable computational cost to achieve high reconstruction accuracy.

Assume that a source and detector assembly is used to sense transmittance at each linear increment along the common z axis of superimposed 3-D Cartesian and polar coordinate spaces. Given a two-dimensional domain $\mathbf{X}$, one thus obtains a collection of images $\mathbf{a} \equiv \{(i, \mathbf{a}_i) : \mathbf{a}_i \in \mathbf{R}^{\mathbf{X}}\}$. Given incident beam intensity I_0, $\mathbf{a}$ can be transformed to obtain an in-plane density function

$$\mathbf{d}(\rho, \theta) = \frac{I_0}{\mathbf{a}(\rho, \theta)}, \quad \text{where} \quad \rho = \sqrt{x^2 + y^2} \quad \text{and} \quad \theta = \tan^{-1}(y/x) \quad \text{with} \quad (x, y) \in \mathbf{X},$$

and the beam direction forms angle θ with the y-axis. We assume for purposes of convenience that $\mathbf{d}$ has the same set-theoretic form as $\mathbf{a}$ and can be operated upon by the Radon transform to yield the following expression for the projection process:

$$\mathbf{b}(\rho, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbf{d}(\rho, \theta) \cdot \delta(x \cos \theta + y \sin \theta)\, dx\, dy, (x, y) \in \mathbf{X},$$

where δ denotes the Dirac delta function with ρ and θ defined previously.

In the case of CAT scanning, image reconstruction is based upon the similarity properties of the two-dimensional Fourier transform $\mathcal{F} : \mathbf{R}^{\mathbf{X}} \rightarrow \mathbf{R}^{\mathbf{X}}$ Given $\mathbf{c} = \mathcal{F}(\mathbf{d})$ we obtain

$$\mathcal{F}(\mathbf{b}(\rho, \theta)) = \mathbf{c}(\xi, \theta),$$

where $\xi = \sqrt{u^2 + v^2}$ and $(u, v) \in \mathit{domain}(\mathcal{F}(\mathbf{b}))$ Thus, each density function yields a function $\mathbf{c}_i$ that represents a radial cross-section through the 2-D FT of the object. By sampling θ in small steps, the resulting transforms can be interpolated to approximate $\mathbf{c}$, which can be inverse-transformed to yield a further approximation to $\mathbf{b}$. Performed over a range of linear increments along the z-axis, this process yields an approximation to the 3-D X-ray density map of the object.

Assuming (for purposes of brevity) that $\mathbf{c}$ is defined on domain $\mathbf{X}$, the computational efficiency of the FT-based reconstruction algorithm can be increased at the expense of accuracy by employing the *back-projection algorithm*, which is given by

$$\mathbf{c}(x, y) = \int_0^{\pi} \mathbf{b}(x \cos \theta + y \sin \theta,\ \theta)\, d\theta, (x, y) \in \mathbf{X}. \qquad \text{(III)}$$

Each projected density function is projected back (expanded) along the axis of the beam at orientation θ to form a 2-D image that is comprised of 1-D images parallel to the scanning axis. By summing (superposing) all such images over one z-axis bin, an approximation of the cross-sectional view can be obtained at that bin.

In practice, the reconstructed function $\mathbf{c}$ is comprised of $\mathbf{d}$ blurred by a point-spread function of form $\mathbf{f}(x, y) = (x^2 + y^2)^{1/2}$. By highpass filtering each 1-D projection in $\mathbf{b}$ prior to backprojection with a filter whose MTF increases linearly with spatial frequency, one can approximate the inverse Radon transform, from which $\mathbf{d}$ is obtained. Lowpass filtering of the reconstruction $\mathbf{d}$ reduces the appearance of random noise at the expense of resolution.

4. Error Analysis and Physical Model Results

Three types of error are expected to predominate in tomography-based reconstruction algorithms. First, occlusion of the target by opaque or translucent cover segments tends to reduce the information about target reflectance (UV and visible wavelengths) and induced or inherent luminosity (UV or infrared wavelengths). This would locally decrease the signal-to-noise ratio (SNR) at a target point visible in a given view of the object. In the presence of point-spread function (PSF) effects, total occlusion at given target points could yield a small amount of useful information due to channel crosstalk, provided that the PSF could be accurately estimated and deconvolved from the received image. However, this is unlikely in battlefield practice without clearly visible reference objects (e.g., manufactured targets such as vehicles) from which one can estimate edge-spread effects [Sch95,Yan95,Sch96b]. In Section 4.1, we discuss this issue in the context of an information-theoretic model of target occlusion.

A second source of error in target reconstruction from multiple views occurs during projection of one or more pixels of a discrete image from the sensor focal plane to a target point. The discrete focal plane has quantization error that is magnified by the projection process. Additionally, media effects such as scattering and attenuation reduce the certainty with which a point may be located in the focal plane. The combined first-order effects of focal plane quantization and PSF effects yield spatial errors that are projected to the domain of the reconstructed target. For example, in near-nadir viewing at low altitudes, such errors tend to be minimized when paraxial approximations are justified by the sensor configuration. In contrast, given wide-field imaging at long range distance (e.g., a cruise missile viewing a distant target on the edge of a clearing, as the missile approaches above treetop level), slant-path effects due to natural or manmade obscurants can severely degrade the received image. If targets are poorly resolved, then quantization error may predominate. The geometric basis for such effects is summarized in Section 4.2 and [Sch97b].

Target reconstruction algorithms present further problems. For example, tomography based on the Fourier transform (per Section 3.3) customarily requires specially designed filtering or data windowing techniques to maximize SNR and minimize artifacts such as high-frequency ringing or vignetting effects [Nat86]. Additionally, the FT has well-known errors that result from division by small-magnitude coefficients, aliasing of high frequency information, etc. The backprojection algorithm given in Equation (III) has associated errors of approximation, as well as computational errors inherent in the transcendental functions, which tend to be dominated by the key error of angular quantization binsize. In fairness, we note that such errors are not absent from geometric reconstruction algorithms, for example, the relative geometry approach of Shashua and Navab [Sha96]. A key problem in geometry-based reconstruction is accurate location of the viewpoint and target reference points. Also, solutions to the correspondence problem are required that have error approaching the Nyquist limit. Errors involved in such processes are discussed extensively in [Nat86, Yes 89, Smi96] and [Sch97b].

Section 4.3 contains a summary of a computer simulation model of MLI processes that was developed during the study and is being enhanced. An additional topic investigated in this study is overviewed in Section 4.4, namely, physical models that can be employed in place of the computer model for fast MLI simulation using detailed target and background environments. The physical modeling methods, which have long been known, suffer from lack of realism at high resolution, but tend to produce useful imagery for early algorithm testing. Furthermore, the model configuration can be easily varied, and results can be verified directly by physical measurement of the sensor parameters.

4.1. Information-Theoretic Model of Target Occlusion

In this section, we generalize the model developed in Section 3 to produce an estimate of information contained in a given projection of an obscured target viewed through perforated cover.

Definition 2. If a process P has n equiprobable outcomes, then the entropy of P is given by $H(\mathrm{P}) = \log(n)$, where the logarithm is taken to the base two.

Definition 3. If a process P has n possible independent outcomes, each of which has probability p_i, then the entropy of P is given by

$$H(\mathrm{P}) = \sum_{i=1}^{n} -\mathrm{p}_i \cdot \log(\mathrm{p}_i) \ .$$

If the outcomes of P are not statistically independent (which occurs in practice), then the equality relation in the preceding equation is replaced with the inequality ($\leq$).

Assumption 3. Let the probability of viewing through cover (per Figure 1) be expressed in terms of an image $\mathbf{c}$ that is comprised of component probability maps $\mathbf{c}_i \in [0, 1]^{\mathbf{X}}$. Let the component images of $\mathbf{c}$ be coregistered on a common domain $\mathbf{Y} \supseteq \mathbf{X}$, such that the n domain points in the target plane to which a given focal-plane point $\mathbf{x} \in \mathbf{X}$ could project correspond to one and only one point in $\mathbf{Y}$. We adopt the convention that, in the i-th view, if a pixel $\mathbf{c}_i(\mathbf{x})$ is zero-valued, then there is no view through the cover at that location. Otherwise, there is partial (total) information transmission from the underlying image if $0 < \mathbf{c}_i(\mathbf{x}) < 1$ ($\mathbf{c}_i(\mathbf{x}) = 1$). Let us conceptualize MLI imagery as comprised of multiple views $\mathbf{a}_i \in \mathbf{F}^{\mathbf{X}}$, where $1 \leq i \leq n$, which are coregistered on $\mathbf{Y} = domain(\mathbf{c})$ to produce an image $\mathbf{b}$ that is comprised of images $\mathbf{b}_i \in \mathbf{F}^{\mathbf{Y}}$, each of which corresponds to a view in $\mathbf{a}$. Let λ denote a flag value and let the operation $f : [0, 1] \rightarrow (0, 1] \cup \{\lambda\}$ map nonzero values to themselves and zero to λ. Further assume that the operation $h : \mathbf{R} \cup \lambda \rightarrow \mathbf{R}$ is a refinement of the logarithm, such that

$$h(x) = \begin{cases} \log(x) & \text{if } x \in (0, 1] \\ 0 & \text{otherwise} \end{cases} \ .$$

Theorem 5. Given Assumption 3, the entropy of $\mathbf{c}$ is given by

$$H(\mathbf{c}(\mathbf{y})) \leq \sum_{i=1}^{n} -\mathbf{c}_i(\mathbf{y}) \cdot h(f[\mathbf{c}_i(\mathbf{y})]) \ .$$

Proof. The proof follows directly from the givens and Definition 3. ■

Assuming, as before, that $\mathbf{b}$ and $\mathbf{c}$ share a common domain $\mathbf{Y}$, then $H(\mathbf{c}(\mathbf{y}))$ portrays the entropy of a pixel $\mathbf{b}(\mathbf{y}) = (\mathbf{b}_1(\mathbf{y}), \mathbf{b}_2(\mathbf{y}), \ldots, \mathbf{b}_i(\mathbf{y}), \ldots, \mathbf{b}_n(\mathbf{y}))$, where $\mathbf{b}_i(\mathbf{y})$ denotes the i-th coregistered view of the $\mathbf{y}$-th target pixel. Observe that this information is useful for target recognition purposes, as discussed in the following algorithms.

Algorithm 2. Given Theorem 5 and the preceding observation, one could exclude the pixels of $\mathbf{c}$ that have no target information via the operation

$$\mathbf{f} = \chi_{=\mathbf{0}}(\mathbf{c}) \ .$$

Algorithm 3. Given Theorem 5 and the preceding observation, it is possible to determine the indices of projections that exceed an entropy threshold T, as follows:

$$\mathrm{D} = \bigcup_{i=1}^{n} domain(\chi_{>\mathrm{T}}(\Sigma\, \mathbf{c}_i)) \ .$$

Thus, one could preselect certain views for likelihood of useful information. This technique also suggests the use of the singular value decomposition (SVD) for determining significant views.

4.2. Projective Errors

From Figure 1, the projection of a point x in a one-dimensional domain **X** to the i-th focal-plane pixel of a sensor's imaging device looking horizontally constitutes the worst case, since the target is in the focal plane periphery. At altitude a_s, focal length f, and magnification M, the projection from a target point $x \in \mathbf{X}$ to the focal plane is given by

$$i = f \cdot \cot\left(M \cdot \left[\frac{\pi}{2} - \tan^{-1}\left(\frac{x}{a_s}\right)\right]\right) .$$

Taking the derivative of i with respect to x, we obtain the worst-case focal-plane error ∂i that results from a location error ∂x in the target plane, as follows:

$$\frac{\partial i}{\partial x} = \frac{f\left(1 + \cot\left(\frac{M}{2}\left[2 \cdot \tan^{-1}\left(\frac{x}{a_s}\right) - \pi\right]\right)^2\right)}{a_s^2 + x^2} .$$

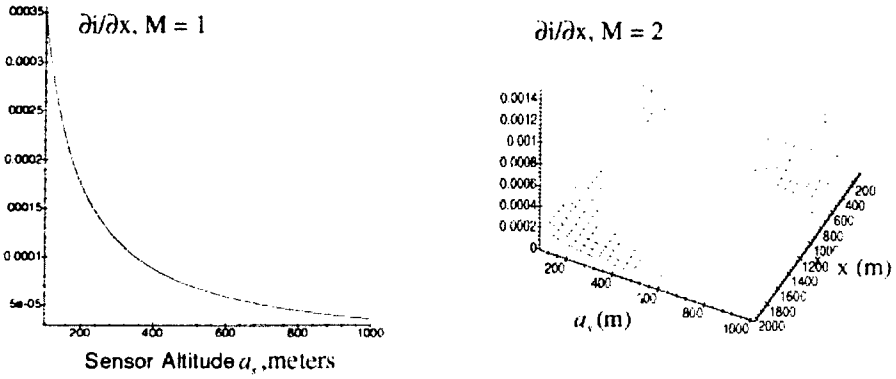


Figure 5. Worst-case target-to-camera projection error (a) M = 1 and (b) M = 2.

Example. Since f and M are usually fixed, we set f = 35mm and M = 1, which are reasonable implementational values. Varying the sensor altitude from 0.1km to 1 km and varying the range distance x from 100m (nadir viewing) to 2km (slant-path viewing), we obtain the result shown in Figure 5a, where $\partial i/\partial x = 0.035/a_s$. Setting M = 2, we obtain the result of Figure 5b, which is described by the following equation:

$$\left.\frac{\partial i}{\partial x}\right|_{M=2} = 0.07 \cdot \frac{a_s \cdot \left(1 + \cot\left(2 \cdot \tan^{-1}\left(\frac{x}{a_s}\right)\right)^2\right)}{a_s^2 + x^2} .$$

The minimum error scenario occurs when the target is on-axis, i.e., when the sensor's optical axis coincides with the line segment between x and a_s. Such error is described by the expression

$$\partial i = f \cdot \tan\left(M\left[\tan^{-1}\left(\frac{x}{a_s}\right) - \tan^{-1}\left(\frac{x + \partial x}{a_s}\right)\right]\right) ,$$

which reduces to $\partial i \approx \partial(x \cdot f/(Mx))$ in the paraxial assumption.

Example. In the minimum-error case for M = 1, the paraxial approximation at a_s = 400m, x = 1km ground range, and $\partial x = \pm 1m$ ground error yields a focal-plane error of

$$\partial i \approx \partial(x \cdot f/(Mx)) = \pm 35\mu = \pm 1m \cdot 1 \cdot 0.035m/10^3 m .$$

Given 9 micron pixels, $\partial i = \pm 35\mu/9\mu = \pm 3.9$ pixels. With 20 micron pixels, $\partial i = \pm 35\mu/20\mu = \pm 1.7$ pixels.

Under similar constraints, during target recognition, the projection of a ray bundle that subtends the i-th focal plane pixel to a target point $x \in \mathbf{X}$ is given by

$$x = a_s \cdot \tan\left(\frac{\pi}{2} - \frac{1}{M} \cdot \tan^{-1}\left(\frac{i}{f}\right)\right) .$$

The general expression for ground error as a function of focal-plane error (the derivative of x with respect to i) is given as follows:

$$\frac{\partial x}{\partial i} = -\frac{a_s\left(1 + \cot\left(\frac{\tan^{-1}(i/f)}{M}\right)\right)}{M(f^2 + i^2)} . \qquad \text{(IV)}$$

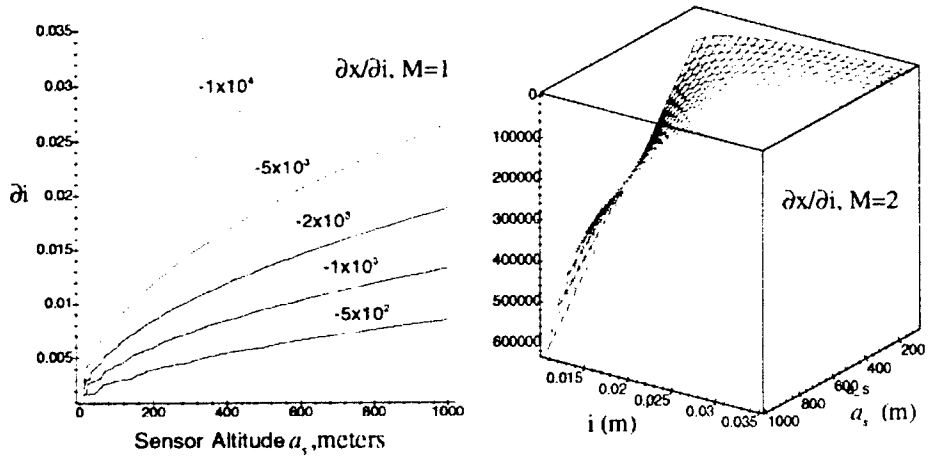


Figure 6. Projection of focal-plane error to the target plane at (a) M = 1 and (b) M = 2.

Setting f = 35mm and M = 1, as before, and varying a_s from 0.1km to 1km and i from 0.01f to 1.0f, we obtain the result shown in Figure 6a. Within the paraxial assumption, the preceding equation reduces to $\partial x \approx \partial i \cdot Mx/f$. Setting M = 2, we obtain the following result shown in Figure 6b:

$$\left.\frac{\partial x}{\partial i}\right|_{M=2} = -\frac{f \cdot a_s \cdot \left(1 + \cot\left(\frac{\tan^{-1}(i/f)}{2}\right)\right)}{2(f^2 + i^2)} .$$

Example. Given the preceding sensor parameters, in the worst case at unitary magnification (Reference Figure 6a), a focal-plane error of $\pm 15\mu$ yields a spatial error in the target plane computed from Equation (IV) as

$$\partial x = \partial i \cdot \left.\frac{\partial x}{\partial i}\right|_{a_s = 400m} = \pm 0.075m = \pm 15\mu \cdot -5 \cdot 10^3 .$$

In practice, we currently estimate that additional effects (e.g., atmospheric PSF, camera noise, reflections, etc.) could increase $\partial x/\partial i$ by a factor of two to three, thereby rendering ∂x sufficiently large to corrupt high-resolution target reconstruction. Although not a significant factor in early research in battlefield MLI (due to assumptions of low spatial resolution), such errors could severely degrade reconstruction of higher-resolution target models.

The preceding error analysis is required for parameterizing image formation and reconstruction processes. In the image formation step, a knowledge of $\partial i/\partial x$ is required due to sensor positioning and aiming errors that are projected to the focal plane. During the selection of a target reconstruction technique, the focal plane error ∂i directly influences the accuracy of target-plane projection, as shown above. More detailed error analyses are given in [Sch97b].

4.3. Computer Model of MLI

We are completing development of a computer model of multi-look imaging that incorporates the information-theoretic model of Section 4.1 with the target projection model of Theorems 3 and 4. Additional features of the computer model are error estimation based on the discussion of Section 4.2 and [Sch97a]. Model architecture is illustrated schematically in Figure 7 and will be discussed in future publications.

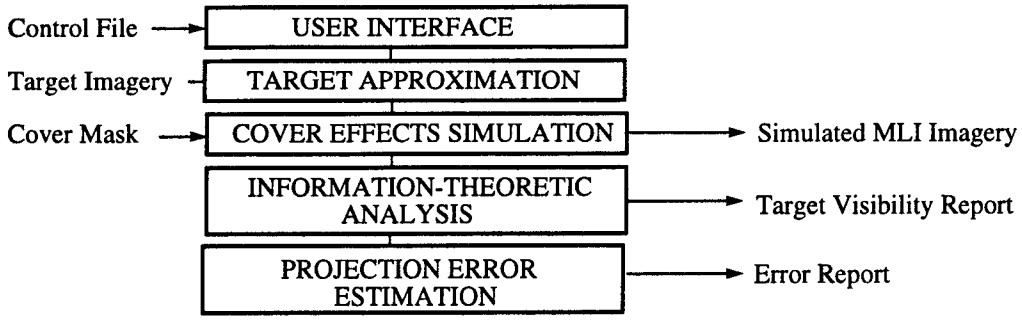


Figure 7. High-level schematic illustration of MLI model architecture.

The target approximation process geometrically warps 2-D images of the target's sides and top to conform to the model of Figure 4a. Cover effects simulation implements the coregistration process discussed in Section 4.1, which yields simulated multi-look imagery of the target obscured by a prespecified 2-D cover mask. This is followed by the information-theoretic analysis highlighted in Section 4.1 and described in detail in [Sch97b], which outputs a target visibility report that can be used to estimate the performance of ATR algorithms applied to the multi-look imagery. The projection error estimation process implements analyses similar to those of Section 4.2 and yields an error report that assesses problems that may occur in given projections due to projection and media errors.

4.4. Physical Models for Simulations of MLI Imaging

As an aside, we note that a less costly alternative to computer generation or field acquisition of MLI imagery is the use of physical models, such as trees, vehicles, and landscaping materials customarily found in hobby scenarios. We have employed this method to inexpensively (< $150 cost) yield realistic MLI imagery, as shown in Figure 8.

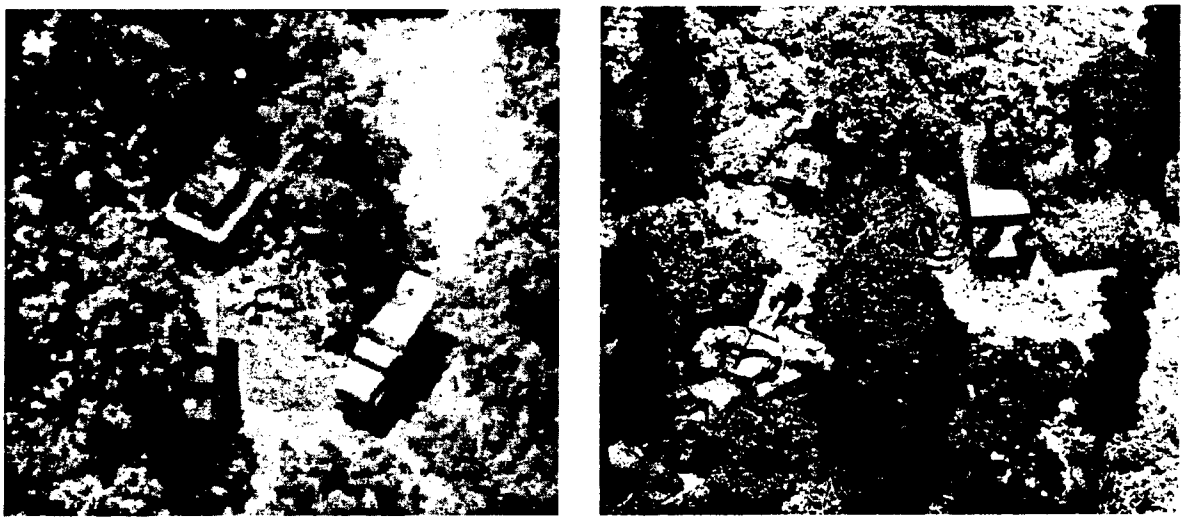


Figure 8. Targets obscured in cover (a) Actual field imagery, and (b) MLI component image of a physical model.

5. Conclusions

Multi-look imaging (MLI) is a new technology for gathering image datasets comprised of multiple views of a given target that is especially useful for finding targets that are partially obscured by clutter. Given MLI data, one could apply target reconstruction algorithms based on relative geometry, stereophotogrammetric or tomographic reconstruction, or coregistration processes, to obtain a two- or three-dimensional approximation to the target. In this study, we conducted an in-depth literature search for MLI algorithms, which were analyzed for noise or error sensitivity. It was found that partial information due to occlusion represented the primary error source, followed by projection effects due to focal plane quantization error and atmospheric phenomena such as scattering and absorption.

A computer model of MLI image formation was constructed that is nearing completion, which would benefit MLI algorithm designers by producing information-theoretic analyses of target visibility and would estimate the affects of systematic, projection, atmospheric, and ground truthing errors on various MLI image formation and target reconstruction processes. Additionally, we found that simple physical models of battlefield scenes could be constructed from model vehicles, foliage, etc., from which MLI imagery can be obtained inexpensively (e.g., less than $150 in this study). A dataset of over 30 high-resolution MLI images was thus acquired, which exemplified on-track, narrow-angle, and loitering modes of low-altitude airborne surveillance. We plan to employ this imagery in a follow-on study of MLI target recognition algorithms. The proposed effort would extend the computer model to include various types of error in target reconstruction processes. This would directly facilitate analysis and implementation of best-performance algorithms for target reconstruction or recognition from MLI imagery.

6. References

[Bar81] Barrow, H.G. and J.M Tenenbaum. "Computational vision", *Proceedings IEEE* **69**(5):572-595 (1981).

[Bas96] Basri, R. and D. Weinshall. "Distance metric between 3D models and 2D images for recognition and classification", *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**(4):465-470 (1996).

[Bes85] Besl, P.J. and R. Jain. "Three-dimensional object recognition", *ACM Computing Surveys* **17**(1):75-145 (1985).

[Bol91] Bolle, R.M. and B.C. Vemuri. "On three-dimensional surface reconstruction methods", *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13**(1):1-13 (1991).

[Chr96] Christy, S. and R. Horaud. "Euclidean shape and motion from multiple perspective views by affine iterations", *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**(11):1098-1104 (1996).

[Der94] Deriche, R., Z. Zhang, Q.T. Luong, and O.D. Faugeras. "Robust recovery of the epipolar geometry for an uncalibrated stereo rig", *Proceedings of the European Conference on Computer Vision, Stockholm, Sweden, May 1994,* pp. 567-576 (1994).

[Fau92] Faugeras, O.D. "What can be seen in three dimensions with an uncalibrated stereo rig?", *Proceedings of the European Conference on Computer Vision, Santa Margherita, Italy, June 1992,* pp. 563-578 (1992).

[Fau96] Rothwell, C. Csurka, G. and O. Faugeras. "Comparison of projective reconstruction methods for pairs of views", *IEEE 1995 International Conference on Computer Vision*, pp. 932-937 (1996).

[Fis81] Fischler, M.A. and R.C. Bolles. "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography", *Communications of the ACM* **24**:381-395 (1981).

[Gri81] Grimson, W.E.L. "Computing surface shape using a theory of human stereo vision", *Proceedings - IEEE Computer Society's 5th International Computer Software & Applications Conference, COMPSAC 81, Chicago, IL*, pp. 209-210 (1981).

[Gri92] Grimson, W.E.L., D.P. Huttenlocher, and T.D. Alter. "Recognizing 3-D objects from 2-D images: An error analysis", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Urbana, IL, USA (1992).

[Har86] Harris, J.G. "The coupled depth-slope approach to surface reconstruction", M.S. Thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, Cambridge, MA (1986).

[Hut90] Huttenlocher and S. Ullman. "Recognizing solid objects by alignment with an image", *International Journal of Computer Vision*, **5**(2):195-212 (1990).

[Lon81] Longuet-Higgins, H.C. "A computer algorithm for reconstructing a scene from two projections", *Nature* **293**:133-135 (1981).

[Lud94] Ludwig, K., J. Neumann, and B. Neumann. "Local stereoscopic depth estimation", *Image and Vision Computing* **12**(1):16-35 (1994).

[Mar78] Marr, D. and H.K. Nishihara. "Representation and recognition of the spatial organization of three dimensional structure", *Proceedings of the Royal Society of London B* **200**:269-294 (1978).

[Mar79] Marr, D. and T. Poggio. "A theory of human stereo vision", *Proceedings of the Royal Society of London B* **204**:301-328 (1979).

[Mei79] Meinguet. "Multivariate interpolation at arbitrary points made simple", *Journal of Applied Mathematics and Physics* **30**:292-304 (1979).

[Min75] Minsky, M. *The Psychology of Computer Vision*, Cambridge: MIT Press (1975).

[Moo96] Moons, T., L. Van Gool, M. Proesmans, and E. Pauwels. "Affine reconstruction from perspective image pairs with a relative object-camera translation in between", *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**(1):77-83 (1996).

[Nat86] Natterer, F. *The Mathematics of Computerized Tomography*, New York: Wiley (1986).

[PAR97] PAR Government Systems Corporation, Inc. "Multispectral Sensor Technology for Automated Target Recognition", Proposal submitted to Rome Laboratory/IRRE (1997).

[Rit96] Ritter, G.X. and J.N. Wilson. *Handbook of Computer Vision Algorithms in Image Algebra*, Boca Raton, FL: CRC Press (1996).

[Sch77] Schumaker, L.L. "Fitting surfaces to scattered data", in *Approximation Theory II*, G.G> Lorentz, C.K. Chui, and L.L. Schumaker, Eds. New York: Academic Press, pp203-267 (1979).

[Sch95] Schmalz, M.S., G.X. Ritter, C.T. Yang, and W.C. Hu. "Center-surround filters for the detection of small targets in cluttered multispectral imagery. 2. Analysis of errors and filter performance", *Proceedings SPIE* **2496**:756-766 (1995).

[Sch96a] Schmalz, M.S. "Advances in predicting spatial deformations in imagery acquired by viewing through the sea surface", *Proceedings SPIE* **2765**:526-537 (1996).

[Sch96b] Schmalz, M.S., G.X. Ritter, and F.M. Caimi. "Data compression techniques for underwater imagery", in *Proceedings IEEE OCEANS '96 Conference*, Ft. Lauderdale, FL (1996).

[Sch97a] Schmalz, M.S. "Automated analysis and prediction of accuracy and performance in ATR algorithms. 2. Experimental results and system performance analysis", in *Proceedings SPIE* **3079** (1996).

[Sch97b] Schmalz, M.S. "MLI - Multi-Look Imaging: Theory and Error Analysis", Unpublished Research Notes, work performed at Rome Laboratory (1997).

[Sha96] Shashua, A. and N. Navab. "Relative affine structure: Canonical model for 3D from 2D geometry and applications", *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**(9):873-883 (1996).

[Smi96] Smith, P.W. and N. Nadhakymar. "An improved power cepstrum based stereo correspondence method for textured scenes", *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**(3):338-348 (1996).

[Ter83] Terzopoulos, D. "Multilevel computational processes for visible surface reconstruction", *Computer Vision, Graphics, and Image Processing* **24**:52-96 (1983).

[Ter88] Terzopoulos, D. "The computation of visible-surface representation", *IEEE Transactions on Pattern Analysis and Machine Intelligence* **10**(4):417-438 (1988).

[Ull91] Ullman, S. and R. Basri. "Recognition by linear combinations of models", *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13**:922-1006 (1991).

[Vem86] Vemuri, B.C., A. Mitiche, and J.K. Aggarwal. "Curvature-based representation of objects from range data", *Image and Vision Computing* **4**(2):107-114 (1986).

[Vem87] Vemuri, B.C. "Representation and recognition of objects from dense range maps", Ph.D. Dissertation, University of Texas at Austin, Department of Electrical and Computer Engineering, Austin, TX (1987).

[Yan95] Yang, C.T., W.C. Hu, M.S. Schmalz, and G.X. Ritter. "Center-surround filters for the detection of small targets in cluttered multispectral imagery. 1. Background and filter design", *Proceedings SPIE* **2496**:756-766 (1995).

[Yes89] Yeshurun, Y. and E.L. Schwartz. "Cepstral filtering on a columnar image architecture: A fast algorithm for binocular stereo segmentation", *IEEE Transactions on Pattern Analysis and Machine Intelligence* **11**:759-767 (1989).

SIMPLE REAL-TIME TRACKING INDICATORS FOR A FREQUENCY FEEDBACK DEMODULATOR

John L. Stensby
Professor
Department of Electrical Engineering

University of Alabama in Huntsville
Huntsville, AL

Final Report for:
Summer Faculty Research Program
Rome Laboratory

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Rome Laboratory

August 1997

SIMPLE REAL-TIME TRACKING INDICATORS FOR A FREQUENCY FEEDBACK DEMODULATOR

John Stensby
Professor
Department of Electrical and Computer Engineering
University of Alabama in Huntsville

Abstract

Frequency feedback demodulators are discussed in general qualitative terms, and a recently proposed feedback demodulator is introduced. Then, two real-time algorithms are presented for assessing the tracking performance of the newly-proposed feedback demodulator. The first tracking indicator uses the fact that the output of the numerically controlled oscillator (NCO) should be highly correlated with a time-aligned version of the input signal when the loop is tracking well. The second tracking indicator employs the input and output signals of the bandpass filter (BPF) in the demodulator. The average power in each of these signals is approximated, and the tracking indicator makes its decision by comparing the ratio of these average powers to a user-specified threshold.

SIMPLE REAL-TIME TRACKING INDICATORS FOR A FREQUENCY FEEDBACK DEMODULATOR

John L. Stensby

I. Introduction

The demodulation of an angle-modulated signal embedded in noise is subjected to a threshold phenomena. Basically, the demodulator performs well, and a good estimate of the message signal is recovered, if the input signal to noise ratio (SNR) is above the demodulator's threshold. More precisely, for above threshold operation, the demodulator's output SNR is a linear function of its input SNR. On the other hand, below-threshold operation is characterized by a nonlinear relationship between input and output SNR; below threshold, the output SNR degrades rapidly with decreasing input SNR. Hence, in practical applications, a demodulator's threshold is an important measure of relative performance.

Advanced angle demodulators can be constructed that employ frequency feedback techniques [1, 2]. In wideband angle modulation applications (where the transmission bandwidth is several times the information bandwidth), these demodulators can have a threshold that is significantly lower than demodulators that do not utilize frequency feedback. That is, in wideband applications, frequency feedback can be used to lower the threshold of an angle demodulator.

The basic idea common to all frequency feedback schemes can be discussed qualitatively. This is accomplished here by utilizing an analog demodulator; however, the ideas and reasoning given in this section apply equally well to digitally-implemented frequency feedback demodulators. On Figure 1, note that message m(t) frequency modulates the input signal. Also, $\eta_{bp}(t)$ represents an input noise disturbance with a bandwidth equal to, or larger than, that of the message-dependent component of the input signal. For proper demodulation, the instantaneous frequency excursion $d\psi_{lo}/dt$ of the voltage controlled oscillator (VCO) must track closely the instantaneous frequency excursion $y(t) = K_D m(t)$ of the input signal. If this can be achieved, then the desired signal component in $x_{bp}(t)$ will be angle modulated with a modulation index that is significantly less than the index of the demodulator input signal. Equivalently, the desired signal component in $x_{bp}(t)$ will have a bandwidth that is significantly less than the bandwidth of the message-dependent component of the loop input signal. However, the down conversion process causes little change in SNR; the SNR calculated for x_{bp} is practically the same as the SNR for the input signal (assuming that the VCO output is approximately uncorrelated with the input noise η_{bp}). Now,

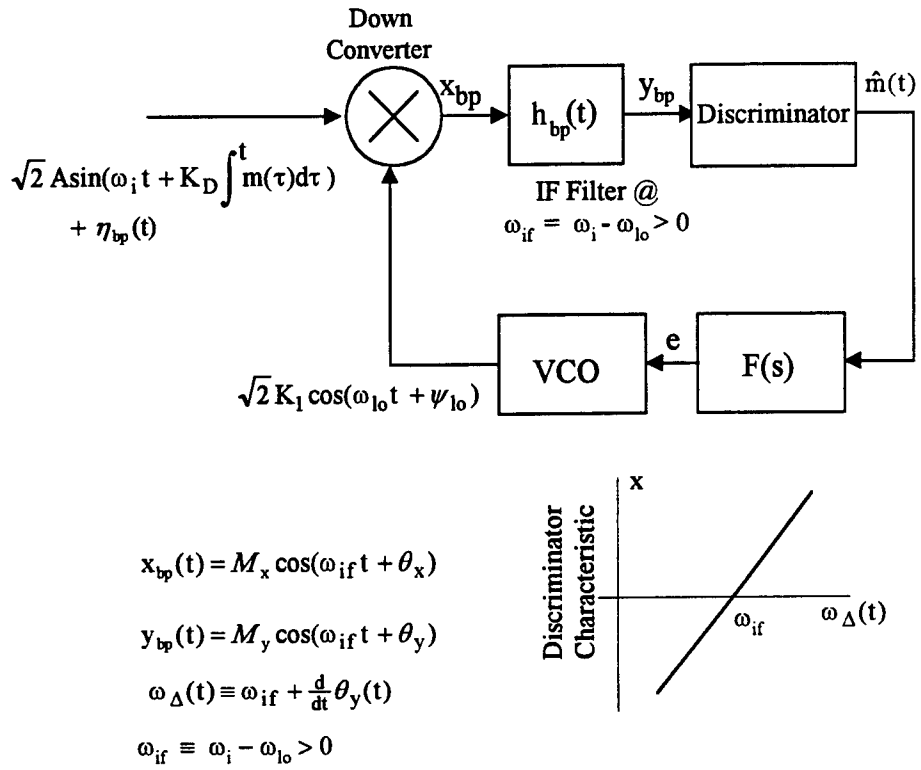


Fig. 1: A simplified frequency feedback demodulator

the loop's IF filter is just wide enough to pass the desired signal component in x_{bp}, and it rejects the bulk of the noise. The IF filter output y_{bp} contains the same desired signal component as x_{bp}, but the former signal has a higher SNR than the latter. Hence, the SNR on the discriminator input is higher (and, presumably, above threshold) than if the loop input signal and noise were applied directly to the discriminator (where a below-threshold operation may be encountered).

There is no real-time indicator of tracking performance in the feedback loop discussed above. The operator of the demodulation loop has little or no sense (either qualitative or quantitative) of loop tracking errors. He may arrive at a judgment by examining the quality of the (recovered) message estimate $\hat{m}(t)$. However, this method may not be practical since it requires *a-priori* knowledge of the transmitted message, information that may not be available to the operator.

This report discusses two methods for generating tracking performance information for a recently proposed digital frequency feedback demodulator. First, in Section II, an overview is given of the frequency

feedback demodulator. Then, Section III describes a tracking indicator that utilizes the fact that the output of the numerically controlled oscillator (NCO) should be highly correlated with a time-aligned version of the input signal when demodulator is operating properly. Finally, Section IV describes a tracking indicator that employs the ratio of output power to input power of the bandpass filter (BPF).

II. Reconstituted Numerical Frequency Modulation Feedback Demodulator

Figure 2 depicts the Reconstituted Numerical Frequency Modulation Feedback Demodulator (RNFMFB), a recently proposed feedback demodulator [2]. It accepts signal $X_+(nT_s;f_0)$, an input bandpass signal expressed in complex form (generally, X_+ consists of an angle modulated signal in additive noise). The demodulator produces the signals $\hat{y}(nT_s) \approx K_D \hat{m}(nT_s)$ and $\hat{\alpha}(nT_s)$. The first of these is an estimate of the message used to frequency modulate input signal X_+. As discussed below, after time alignment by n_0 clock periods (an integer equal to one

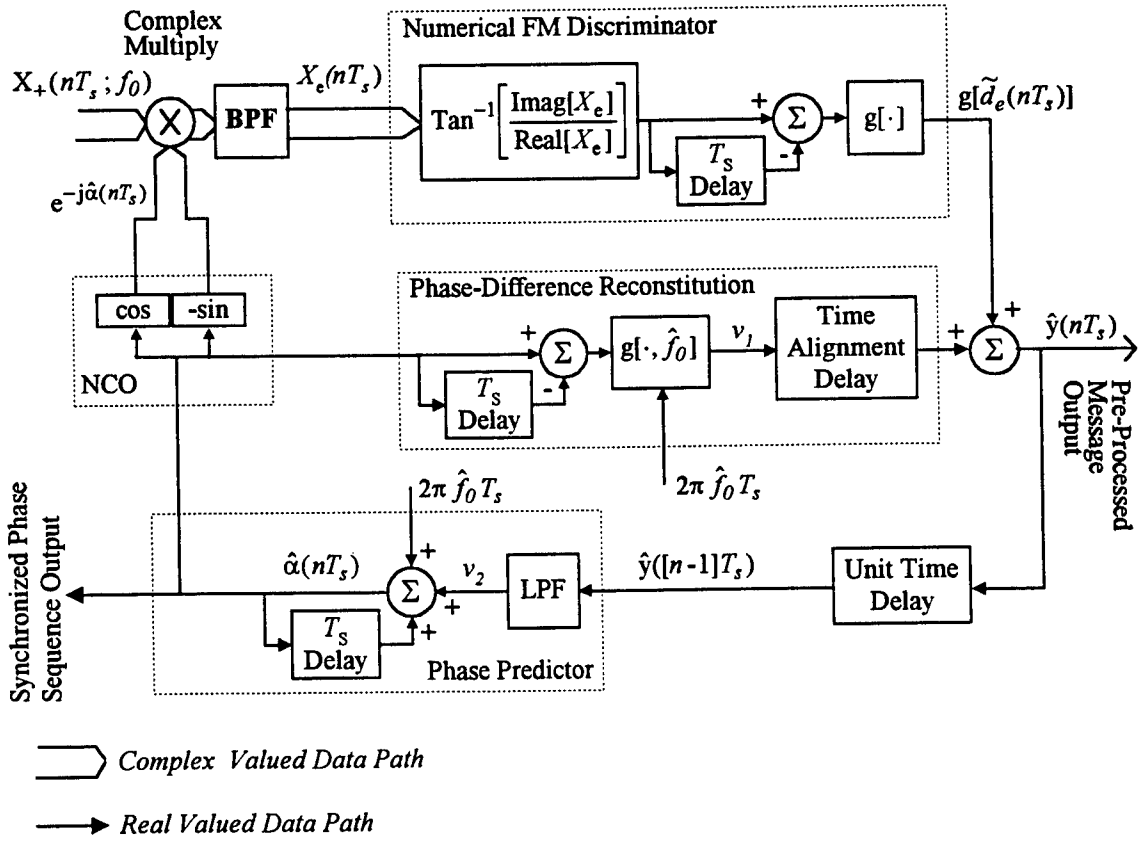


Fig 2: RNFMFB demodulator.

plus the sum of the filter group delays), signal $X_+([n - n_0]T_s; f_0)$ should be highly correlated with $\exp[\hat{\alpha}(nT_s)]$ if the feedback loop demodulator is tracking well. Finally, the RNFMFB is described by the set of state equations given in Appendix II.

Except for the function g and the Time Alignment Delay, all of the function blocks on Figure 2 are self explanatory. Function $g[x] = \mathrm{mod}_{2\pi}(x)$ denotes the modulo 2π value of x, and this function is defined in Appendix I. As used on Figure 2, the function $g[x; \hat{f}_0]$ is defined as

$$g[x; \hat{f}_0] = g[x - 2\pi \hat{f}_0 T_s]. \tag{II-1}$$

The Time Alignment Delay function block provides a user-specified delay of n_d clock periods, a value nominally set to the group delay of the BPF. Finally, constant $\hat{f}_0$ represents a user-specified local estimate of the input carrier frequency.

In the work described here, the input signal is given a special form. For simulation purposes, the input of the demodulator can be expressed as

$$\begin{aligned} X_+(nT_s; f_0) &= A\exp(j\alpha(nT_s)) + (\eta_i(nT_s) + j\eta_q(nT_s))\exp(j2\pi f_0 nT_s) \\ &= A\exp(j[2\pi f_0 nT_s + \alpha_r(nT_s)]) + (\eta_i(nT_s) + j\eta_q(nT_s))\exp(j2\pi f_0 nT_s), \end{aligned} \tag{II-2}$$

where

$$\alpha_r(nT_s) \equiv \alpha(nT_s) - 2\pi f_0 nT_s \tag{II-3}$$

is the *relative* phase of the desired input signal. The desired signal component in (II-2) is a bandpass process with a constant amplitude A, a carrier frequency of f_0 and a message-dependent angle $\alpha_r(nT_s)$. Without loss of generality, it is possible to replace f_0 in (II-2) by $\hat{f}_0$ and include any *detuning error* $2\pi(f_0 - \hat{f}_0)nT_s$ in the definitions of α_r, η_i and η_q. This simplification is for convenience; it reduces the number of variables, and it is employed in what follows. The noise component in (II-2) is a Gaussian bandpass process; $\eta_i(m)$ and $\eta_q(m)$ are zero mean,

stationary Gaussian random processes with

$$E[\eta_i(lT_s)\eta_i(mT_s)] = E[\eta_q(lT_s)\eta_q(mT_s)] = \sigma^2\delta_{lm}$$
$$E[\eta_i(lT_s)\eta_q(mT_s)] = 0 \qquad \text{(II-4)}$$

for all integers l and m. In the first of these formulas, δ_{lm} is the commonly-used unit sample function defined by

$$\delta_{lm} = 1, \quad l = m$$
$$= 0, \quad l \neq m \qquad \text{(II-5)}$$

A simple relationship between v_1 and v_2 can be obtained. From Figure 2 and (II-1), the relationships

$$\hat{\alpha}(nT_s) = \hat{\alpha}([n-1]T_s) + 2\pi\hat{f}_0T_s + v_2(nT_s)$$
$$v_1(nT_s) = g(\hat{\alpha}(nT_s) - \hat{\alpha}([n-1]T_s) - 2\pi\hat{f}_0T_s) \qquad \text{(II-6)}$$

are obtained by inspection. Now, these last two equations can be combined to obtain the desired relationship

$$v_1(nT_s) = g(v_2(nT_s)). \qquad \text{(II-7)}$$

To understand the phase tracking abilities of the feedback demodulator, consider first the relatively simple case when the BPF and the LPF are removed from the loop (*i.e.*, set their transfer functions to unity), there is no Time Alignment Delay function, and the input noise is set to zero. Under these conditions the output of the Numerical FM Discriminator becomes

$$g\left(\tilde{d}_e(nT_s)\right) = g([\alpha(nT_s) - \hat{\alpha}(nT_s)] - [\alpha([n-1]T_s) - \hat{\alpha}([n-1]T_s)])$$
$$= g([\alpha(nT_s) - \alpha([n-1]T_s)] - [\hat{\alpha}(nT_s) - \hat{\alpha}([n-1]T_s)]), \qquad \text{(II-8)}$$

and the input to the Phase Predictor block is

$$v_2(nT_s) = \hat{y}([n-1]T_s)\,. \tag{II-9}$$

From Figure 2, (II-6) and (II-7), the difference equation

$$\hat{y}(nT_s) = g\big(\hat{y}([n-1]T_s)\big) + g\big([\alpha(nT_s) - \alpha([n-1]T_s)] - [\hat{\alpha}(nT_s) - \hat{\alpha}([n-1]T_s)]\big) \tag{II-10}$$

can be obtained easily. Finally, combine (II-5), (II-9) and (II-10) to write

$$\begin{aligned} \hat{\alpha}_r(nT_s) &= \hat{\alpha}_r([n-1]T_s) + \hat{y}([n-1]T_s) \\ \hat{y}(nT_s) &= g\big(\hat{y}([n-1]T_s)\big) + g\big([\alpha_r(nT_s) - \alpha_r([n-1]T_s)] - \hat{y}([n-1]T_s)\big), \end{aligned} \tag{II-11}$$

where

$$\hat{\alpha}_r(nT_s) \equiv \hat{\alpha}(nT_s) - 2\pi f_0 nT_s \tag{II-12}$$

is the *relative* phase of the NCO. Equation (II-11) is a time-invariant and nonlinear set of difference equations that describe the loop with no BPF, LPF or Time Alignment Delay functions.

The phase tracking abilities of the demodulator loop are of interest. To examine these abilities, combine the two equations in (II-11) to obtain

$$\hat{\alpha}_r(nT_s) - \hat{\alpha}_r([n-1]T_s) = g\big(\hat{y}([n-2]T_s)\big) + g\big([\alpha_r([n-1]T_s) - \alpha_r([n-2]T_s)] - \hat{y}([n-2]T_s)\big). \tag{II-13}$$

Inspection of this last equation reveals that

$$\hat{\alpha}_r(nT_s) - \hat{\alpha}_r([n-1]T_s) = \alpha_r([n-1]T_s) - \alpha_r([n-2]T_s) \pmod{2\pi}, \tag{II-14}$$

where equality is in the modulo-2π sense. Now, sum both sides of (II-14) to obtain

$$\hat{\alpha}_r(nT_s) - \hat{\alpha}_r(0) = \alpha_r([n-1]T_s) - \alpha_r(-1) \qquad (\text{mod } 2\pi). \qquad \text{(II-15)}$$

Under the initial conditions $\hat{\alpha}_r(0) = \alpha_r(-1)$ the result

$$\hat{\alpha}_r(nT_s) = \alpha_r([n-1]T_s) \qquad (\text{mod } 2\pi) \qquad \text{(II-16)}$$

follows. This analysis indicates that, up to an additive constant, the phase of the NCO is equal to a time-aligned (*i.e.*, delayed) version of the input phase (Note that arbitrary constants can be added to $\hat{\alpha}_r$ without violating (II-14).). As discussed in Section III, this property serves as the basis of a tracking indicator based on correlation.

In the sense described by the analysis leading to (II-16), the phase of the NCO tracks the input phase. However, it must be remembered that the NCO's actual phase is computed only up to an additive, initial-condition dependent, constant. Depending on initial conditions, the phase of the NCO does not asymptotically approach a time-aligned version of the input phase (*i.e.*, constant phase errors do not "naturally die out"). Hence, in the traditional sense associated with phase-locked loops, the NCO is not phase locked to the input reference.

In general, the demodulator loop contains a bandpass filter (the BPF block) and a lowpass filter (the LPF block). In the work described here, the BPF was implemented by two identical lowpass filters, one filter each for the real and imaginary parts (*i.e.*, in-phase and quadrature components) of the complex multiplier output. In this report, these functions are modeled using conventional state space techniques. The BPF is modeled by

$$\vec{X}(nT_s) = \mathbf{A}\vec{X}([n-1]T_s) + \mathbf{B}v_{in}([n-1]T_s)$$

$$v_{out}(nT_s) = \mathbf{C}\vec{X}(nT_s) + \mathbf{D}v_{in}(nT_s)$$

$\mathbf{A}$ is an $m_1 \times m_1$ constant matrix

$\mathbf{B}$ is an $m_1 \times 1$ constant matrix (II-17)

$\mathbf{C}$ is an $1 \times m_1$ constant matrix

$\mathbf{D}$ is a scalar,

where the input and output signals are denoted as $v_{in}(n)$ and $v_{out}(n)$, respectively. In a similar manner, the LPF is modeled by

$$\vec{Y}(nT_s) = \mathbf{A}_L\vec{Y}([n-1]T_s) + \mathbf{B}_L u_{in}([n-1]T_s)$$

$$u_{out}(nT_s) = \mathbf{C}_L\vec{Y}(nT_s) + \mathbf{D}_L u_{in}(nT_s)$$

$\mathbf{A}_L$ is an $m_2 \times m_2$ constant matrix

$\mathbf{B}_L$ is an $m_2 \times 1$ constant matrix (II-18)

$\mathbf{C}_L$ is an $1 \times m_2$ constant matrix

$\mathbf{D}_L$ is a scalar,

where the input and output signals are denoted as $u_{in}(n)$ and $u_{out}(n)$, respectively.

III. Correlation-Based Tracking Indicator

The simple case involving no bandpass or lowpass filters, no Time Alignment Delay function and no input noise was considered in the previous section. Under these conditions, Equations (II-2) and (II-16) show that

$$\overline{\left| X_+([n-1]T_s; \hat{f}_0)\exp[-j(\hat{\alpha}_r(nT_s) + 2\pi f_0 nT_s)] \right|} = A, \tag{III-1}$$

where the over bar denotes a time average. For this trivial case, up to the scale factor A, the NCO output reproduces a time-aligned version of the input signal.

Under certain conditions, a similar situation can be expected when the BPF and LPF are in place. Consider the case when the filters (*i.e.*, the BPF and LPF) have amplitude and phase responses that are flat and linear, respectively, over the bandwidth of their input signals. Under this condition, the filters introduce no amplitude or phase distortion; instead, each filter passes to its output a signal that is a time delayed (by the group delay of the filter) replica of its input. More importantly, up to the scale factor A, the NCO output *approximates* a time-aligned version of the input signal so that

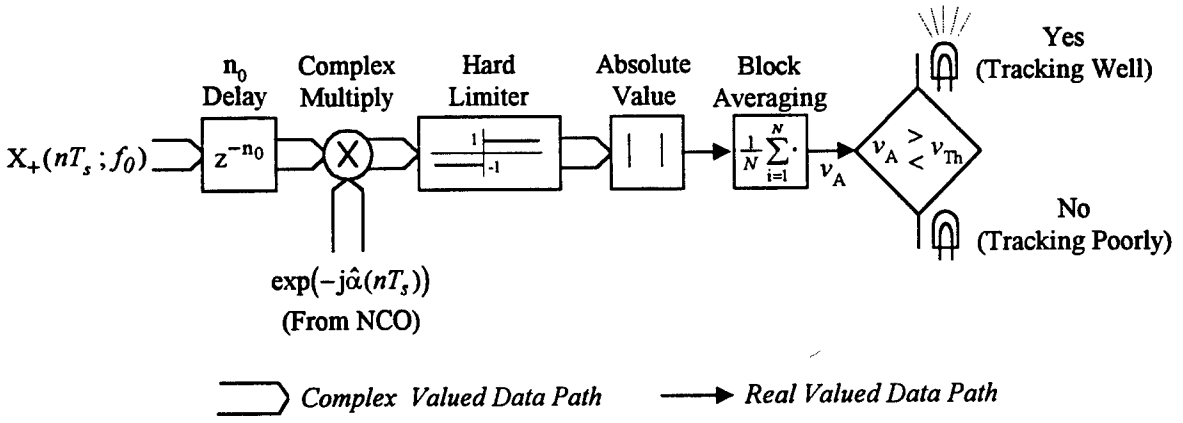


Fig. 3: Correlation-based tracking indicator.

$$\left| X_+([n-n_0]T_s;\hat{f}_0)\exp[-j\{\hat{\alpha}_r(nT_s)+2\pi\hat{f}_0nT_s\}] \right| \approx A\,, \qquad \text{(III-2)}$$

where integer n_0 is one plus the sum of the BPF and LPF group delays. Finally, when noise is added to the input X_+, the time average in (III-2) should be replaced by a probabilistic average (*i.e.*, expectation). In many practical applications, approximation (III-2) will be useable until excessive phase error variance and noise-induced cycle slips render the demodulator useless.

Figure 3 illustrates a practical tracking indicator that utilizes the ideas outlined above. First, the product of the time-aligned input is hard limited to help remove amplitude fluctuations in the received signal. The complex valued nature of the signal is retained since the hard limiting is applied separately to the in-phase and quadrature components of the signal. It should be noted that this step may not be necessary if the amplitude of X_+ is regulated by an amplitude gain control (AGC) subsystem (or hard limiting) prior to the tracking indicator. Next, the absolute value is taken of the hard limited signal, and this real-valued, non-negative result is block averaged (non overlapping blocks of length N are processed). Finally, each block average is compared with a user-supplied threshold v_{Th} to make a tracking/non-tracking decision. In applications, threshold v_{Th} should be based on a tolerance for incorrect decisions.

Figure 4 depicts numerical results that are typical for this tracking indicator. For this simulation, the demodulator employed a BPF comprised of two identical linear-phase, 16-order-FIR, lowpass filters (one each for

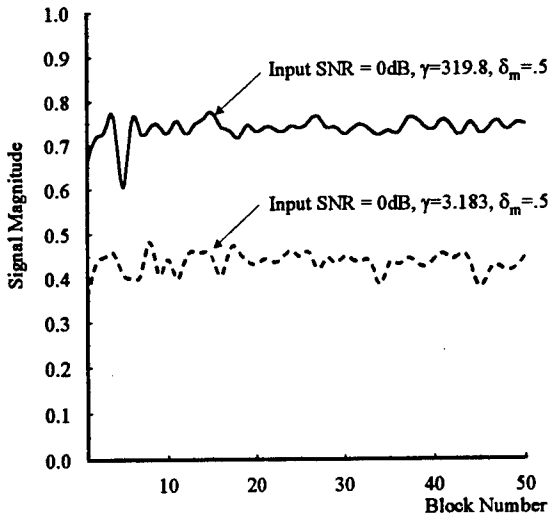


Fig. 4: Average of non-overlapping blocks (N = 1000) of data generated by the Correlation Based Tracking Indicator.

the in-phase and quadrature components of the processed signal) designed to have a cut off frequency of $\pi/10$ radians/second (filter design by the MatLab routine *FIR1* [3]). In the demodulator, the LPF block was implemented by a linear-phase, 8-order-FIR, lowpass filter designed to have a cut off frequency of $\pi/10$ radians/second (filter design by the MatLab routine *FIR1* [3]). The results depicted here are for an input angle modulation given by

$$\alpha_r(nT_s) = \gamma\delta_m \sin\left(\frac{\pi}{\gamma}n\right), \tag{III-3}$$

where γ and δ_m are modulation parameters described by Noga [2]. Also, the input signal and noise powers were both set at one watt (input SNR = 0 dB).

The solid line graph on Figure 4 corresponds to a message frequency of $\pi/\gamma \approx 9.82\times10^{-3}$, a value well within the demodulator bandwidth. Conversely, the dashed line graph corresponds to a message frequency of .987, a value outside of the demodulator bandwidth. Hence, these two examples can be used to illustrate a "working case" and a "non-working case". On Figure 4, an inspection of the plots reveals that these extreme cases can be distinguished if the indicator algorithm utilizes a threshold $v_{Th} \approx .56$, a value midway between the plots.

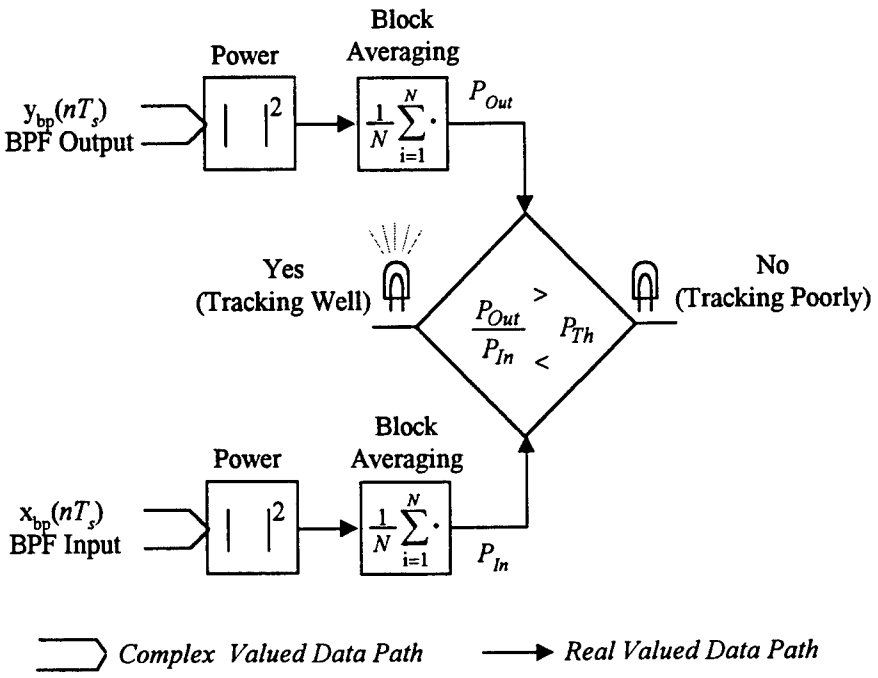


Fig. 5: Tracking Performance Indicator based on average power considerations..

IV. A Tracking Indicator Based on Average Power Considerations

Figure 5 depicts a proposed tracking indicator based on simple ideas. This tracking indicator is supplied with $x_{bp}(nT_s)$ and $y_{bp}(nT_s)$, the input and output, respectively, of the demodulator BPF. First, consider the relatively simple case when the demodulator and BPF inputs are pure noise. For large block size N, the ratio P_{Out}/P_{In} (defined on Fig. 5) should approximate B_W, the BPF bandwidth specified as a fraction of the Nyquist rate (B_W is a parameter supplied to the filter design routines in MatLab [3]) . On the other hand, the ratio P_{Out}/P_{In} should approximate unity when the demodulator and BPF inputs contain only desired signal (no noise), and the demodulator is tracking well. When the demodulator is functioning well, the desired signal component in $x_{bp}(nT_s)$ lies within the passband of the BPF, and $P_{Out} > B_W P_{In}$. In applications, threshold P_{th} should be based on a tollerance for incorrect decisions, and it should satisfy $B_W < P_{Th} < 1$.

Figures 6 and 7 depict numerical results that are typical for this tracking indicator with an input signal and noise that are described in Sections II and III. Figure 6 corresponds to $\gamma = 319.8$, $\delta_m = .5$ and input SNR = 0; for this case, the demodulator is tracking adequately a sinusoidal message at frequency $\pi/\gamma \approx 9.82\times10^{-3}$, a value within the passband of the demodulator. Conversely, Figure 7 corresponds to $\gamma = 3.18$, $\delta_m = .5$ and input SNR = 0;

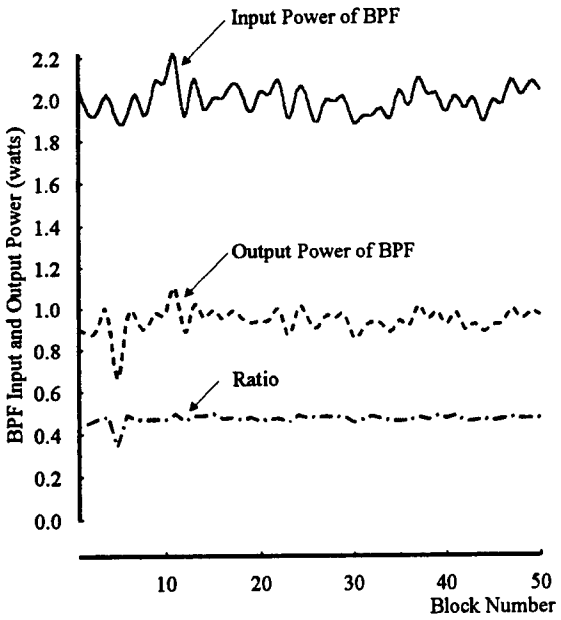


Fig. 6: BPF input and output powers, and their ratio. Plotted for the case $\gamma = 319.8$, $\delta_m = .5$, signal input power = noise input power = 1 watt (input SNR = 0 dB).

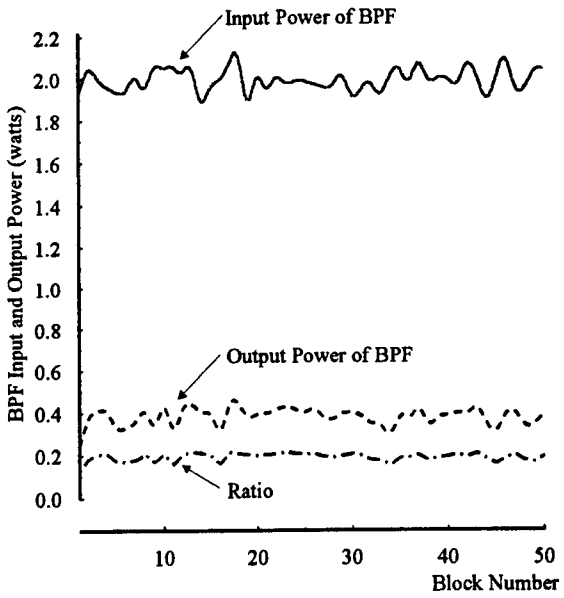


Fig. 7: BPF input and output powers, and their ratio. Plotted for the case $\gamma = 3.18$, $\delta_m = .5$, signal input power = noise input power = 1 watt (input SNR = 0 dB).

the sinusoidal input message frequency is $\pi/\gamma \approx .987$, a value outside of the demodulator bandwidth. Inspection of the P_{Out}/P_{In} ratio plots reveals that these extreme cases can be distinguished if the indicator algorithm utilizes a threshold $P_{th} \approx .35$, a value midway between the ratio plots.

V. Conclusions

The basic ideas common to all frequency feedback demodulation schemes were discussed qualitatively. Then, a recently proposed digital feedback demodulator was introduced, and its operation was summarized. For this demodulator, two tracking indicator algorithms were proposed. The algorithms employ a user-specified threshold to generate a binary decision that indicates "tracking properly" or "not tracking properly". The performance of the algorithms was discussed, and typical numerical simulation results were given.

The first tracking indicator algorithm utilizes the fact that the output of the NCO should be highly correlated with a time-aligned version of the input signal. This algorithm is the counterpart of the commonly-used quadrature lock detector in phase-locked loop theory. The algorithm makes a decision based on the average product of the time-aligned input and NCO signals.

The second tracking indicator algorithm employs the input and output signals of the bandpass filter in the feedback demodulator. The average power in each of these signals is approximated by a block averaging technique. These average power approximations are used in the decision making process; to make a decision, the ratio of these approximations is compared to a user-specified threshold.

This work could not have been completed without the assistance of Dr. Andrew Noga and the staff at Rome Laboratory. Dr. Stensby greatly appreciates the support that he received from Rome Laboratory personnel during this project. Also, he is thankful for the opportunities extended to him by the Air Force Summer Faculty Research Program.

References

[1] Klapper, J. and Frankle, J.T., *Phaselocked and Frequency Feedback Systems*, Academic Press, New York, 1972.

[2] Noga, A., *Performance Analysis and Simulation of a Class of Numerical Demodulation Techniques for Large Deviation FM Signals*, Ph.D. Dissertation, Syracuse University, December, 1995.

[3] Krauss, T.P., Shure, L., Little, J.N., *Signal Processing ToolBox: For Use With MatLab,* The Math Works, Inc., Natick, Mass, 1994.

[4] Stensby, J.L., *Phase-Locked Loops: Theory and Applications*, CRC Press, Boca Raton, FL, 1997.

Appendix I - The Least Residue Function

Given two real numbers x and y, *x is congruent to y modulo* 2π if 2π divides without remainder the difference x - y. Often, this is denoted as

$$x = y \pmod{2\pi}. \tag{A1-1}$$

Note the symmetry here. If $x = y \pmod{2\pi}$, then it is obvious that $y = x \pmod{2\pi}$.

When (A1-1) is true, the quantity y is said to be a *residue* of x. Real number y is said to the *least residue* of x if y is a residue of x **and** $-\pi < y \leq \pi$. Often, the least residue of x is denoted as

$$y = \mathrm{mod}_{2\pi}[x]. \tag{A1-2}$$

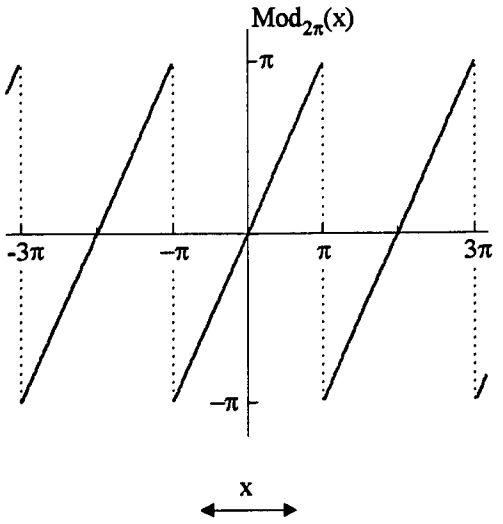


Fig. A1: Phase Characteristic function $Mod_{2\pi}(x)$

In the engineering literature, Function (A1-2) is called the *modulo-2π function.*

The function $y = mod_{2\pi}[x]$ can be computed by adding or subtracting from x integer multiples of 2π. The desired relationship is

$$y = \text{mod}_{2\pi}[x] = \begin{cases} x - 2\pi \cdot \text{Int}\left(+\frac{K}{2}+1\right) & , K\pi < x \le (K+2)\pi \quad , K > 0 \\ x & , -\pi < x \le \pi \\ x + 2\pi \cdot \text{Int}\left(-\frac{K}{2}+1\right) & , (K-2)\pi < x \le K\pi \quad , K < 0 \end{cases} \quad , \qquad \text{(A1-3)}$$

where K is an **odd** integer, and Int[r] denotes the integer part of r (truncate the fractional part).

Figure A1 illustrates a plot of $y = mod_{2\pi}[x]$. Note that the function has step discontinuities at each odd multiple of π. However, from (A1-3), note that the function is left continuous.

Appendix II

State variables for the demodulator must be defined. The BPF has m_1 states, and it is described by an $m_1 \times 1$ vector $\vec{X}(nT_s)$ The LPF has m_2 states; it is described by an $m_2 \times 1$ vector $\vec{Y}(nT_s)$. The instantaneous relative phase of the NCO is described by the single state $\hat{\alpha}_r(nT_s)$. The message output is described by the single

variable $\hat{y}(nT_s)$. Variable Arg(nT_s) describes the phase of the BPF output. Finally, variables $Z_0(nT_s)$, $Z_1(nT_s)$, ... , $Z_{n_d}(nT_s)$ are used in the Time Alignment Function. Processing of the demodulator state is done in a sequential fashion indexed on the time variable n. The equations that describe the state are described in this appendix.

First, the BPF state vector $\vec{X}(nT_s)$ is updated. This is accomplished by computing

$$\vec{X}(nT_s) = \mathbf{A}\vec{X}([n-1]T_s) + \mathbf{B}\Big[\exp\big(j\alpha_r([n-1]T_s) - j\hat{\alpha}_r([n-1]T_s)\big) + \big(\eta_i([n-1]T_s) - j\eta_q([n-1]T_s)\big)\exp\big(-j\hat{\alpha}_r([n-1]T_s)\big)\Big], \quad \text{(A2-1)}$$

where **A** is an $m_1 \times m_1$ constant matrix, and **B** is an $m_1 \times 1$ constant matrix. As described in Section II, α_r is the modulation-dependent relative phase of the input signal.

Next, the NCO relative phase $\hat{\alpha}_r(nT_s)$ is updated. To accomplish this, use the equation

$$\hat{\alpha}_r(nT_s) = \hat{\alpha}_r([n\text{-}1]T_s) + [\mathbf{C}_L\vec{Y}([n\text{-}1]T_s) + \mathbf{D}_L\hat{y}([n\text{-}1]T_s)], \quad \text{(A2-2)}$$

where $\mathbf{C}_L$ is an $m_2 \times m_2$ constant matrix, and $\mathbf{D}_L$ is an $m_2 \times 1$ constant matrix.

The Time Alignment States $Z_0(nT_s)$, $Z_1(nT_s)$, ... , $Z_{n_d}(nT_s)$ must be updated. The relevant equations are

$$\begin{aligned} Z_{n_d}(nT_s) &= Z_{n_d-1}([n-1]T_s) \\ Z_{n_d-1}(nT_s) &= Z_{n_d-2}([n-1]T_s) \\ &\vdots \\ Z_1(nT_s) &= Z_0([n-1]T_s) \\ Z_0([n-1]T_s) &= g\big(\mathbf{C}_L\vec{Y}([n\text{-}1]T_s) + \mathbf{D}_L\hat{y}([n\text{-}1]T_s)\big). \end{aligned} \quad \text{(A2-3)}$$

The LPF state vector $\vec{Y}(nT_s)$ is updated next. This involves the equation

$$\vec{Y}(nT_s) = \mathbf{A}_L\vec{Y}([n-1]T_s) + \mathbf{B}_L\hat{y}([n-1]T_s), \quad \text{(A2-4)}$$

where $\mathbf{A}_L$ is an $m_2 \times m_2$ constant matrix, and $\mathbf{B}_L$ is an $m_2 \times 1$ constant matrix.

Arg(nT_s) describes the phase of the BPF output. To compute this quantity, the complex-valued BPF output must be computed first; this is accomplished by

$$B_{out}(nT_s) = \mathbf{C}\bar{\mathbf{X}}(nT_s) + \mathbf{D}\exp\left(j\alpha_r([n-1]T_s) - j\hat{\alpha}_r([n-1]T_s)\right) + \mathbf{D}\left(\eta_i([n-1]T_s) - j\eta_q([n-1]T_s)\right)\exp\left(-j\hat{\alpha}_r([n-1]T_s)\right) \quad \text{(A2-5)}$$

Then, the BPF angle is computed by using the equation

$$\text{Arg}(nT_s) = Angle\left(B_{out}(nT_s)\right), \quad \text{(A2-6)}$$

where *Angle* is the four-quadrant angle function that returns a value in the range $(-\pi, \pi]$.

Finally, the message output $\hat{y}(nT_s)$ is updated by

$$\hat{y}(nT_s) = Z_{n_d}(nT_s) + g\left(\text{Arg}(nT_s) - \text{Arg}([n-1]T_s)\right). \quad \text{(A2-7)}$$

DESTRUCTIVE-OBJECTS

Michael C. Stinson
Associate Professor
Department of Computer Science and Engineering

Central Michigan University
Mt. Pleasant, MI 48859

Final Report for:
Summer Research Program
Rome Laboratory

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, Washington, DC

And

Rome Laboratory

September 1997

Destructive Objects

Michael C. Stinson
Professor

and

Pat Garrity
Graduate Student

Department of Computer Science
Pearce Hall
Central Michigan University
Mt. Pleasant, MI 48859

Abstract

The use of object and object oriented technology has become one of the most widely used techniques in computer programming. This work examines the structure, development, and use of objects in order to examine the areas for potential run-time errors. This work defines a destructive object in relationship to its programming environment and considers how this type of object might be developed.

The problems that objects can cause in a system are neither well known nor are they simple to understand. This report takes a look at objects and their development in order to understand where undesirable properties of objects may come from. In addition to defining the destructive types of objects, this report looks at how two new approaches to using objects: language dependent and language independent, handle destructive properties. These approaches are considered for their ability to avoid certain destructive object behavior.

Destructive Objects

Michael C. Stinson
Professor

and

Pat Garrity
Graduate Student

Department of Computer Science
Pearce Hall
Central Michigan University
Mt. Pleasant, MI 48859

I. Introduction

In an effort to maintain technical superiority, new technologies in computing have brought about the extensive use of objects, objects oriented programming, and object technology. This movement has been pushed by the need for unified data access and by the improvements in distributed data processing. Objects have the advantages of flexibility and utility not seen before in computer approaches. Moreover, even if objects were used in stand alone systems, they have the advantage of reusability either alone or as a foundation of a more complex object.

Object technologies combine functionality with information in the form of *objects*. This approach could change the way we utilize traditional data in the larger systems. Many of the "smart" or "intelligent" systems are a combination of data and functionality. For example, this multi-faceted approach could be utilized with remote sensors. Traditionally remote sensors collect data, the data is then transmitted and analyzed by a central unit. The analyzed data, now information, can be combined with control information and transmitted to a utilization point. With objects, the data and analytical tools could be used at the sensor site. This would eliminate the additional transmission without jeopardizing the basic structure. Thus calibration, analysis, and applicability could be placed into one package. The sensor information might link directly into a map of a conflict region and could be interpreted based on the command information while the data is simultaneously uploaded.

Due to the complex nature of objects, they bring with them a risk of introducing destructive behaviors, both inadvertent and malevolent, to a system. This report examines the structure, creation, and use of objects to determine where destructive behaviors are introduced.

Object oriented technology has changed our approach to computing as well as the way we view programs. Objects may link together at run-time and execute on a remote server, forgoing the well known and understood concepts of compiling, linking, and running code. The confusion this creates is exacerbated by the nature of objects themselves. Object are, through necessity, more complex than previous constructs. Objects often encapsulate a breadth of functionality and information that is not expected to be used in every instance. This leaves each object with significant capabilities that are not only not used, but are not even known of by the user. Due to the abstract nature of objects it may also be too difficult to run an analysis (finite state machine) on portions of the code to examine the actions that code may take. It is therefore critical that we examine the nature of objects for the possible ill affects that they may add to a system.

II. Background

To clarify the concepts we are discussing this report will consider define objects and their accompanying object classes. It will consider the major components of an objects and how they relate to the process of software development. The report will then define what a destructive process is and why they are detrimental to software.

II.1 Objects and Classes

The basic building blocks in the object oriented paradigm are the *object class* and the *object*. These concepts can best be grasped by looking at an example. If someone talks about maps, it is understood what they are talking about. If they ask "Can you read a map?", you might answer "What kind of map"? This is because a map is a class, an abstract grouping of topological or political information. While you may be able to read a road map or a political map, different skills are required in order to interpret a topographical map. Each one of these categories are classes of maps. They are sub-classes of the superclass: map. Thus we see the hierarchy of classes taking shape. This is how object classes form a hierarchy in their creation.

Traditional data structures hold data with some basic rules about the data. Object classes differ from traditional data structures in that they contain the structure to hold data and the procedures (methods) to process data. Procedures contain the ability to process data without the capability to maintain the data. A software object class creates a repository for data in its *attributes* (variables and constants) and maintains its functionality in its *methods* (functions or procedures).

Definition: An *object class* is a software bundle of attributes (variables and constants) and related methods (procedures).

Objects are constructed (instantiated) from the object class, usually with some additional information. An instantiation of a class must have an *identity* to distinguish it from any other object constructed from that class. For example, if a political map of Europe is created from the class *political map*, it might have the designation *Europe circa 1812*. An object may also have specific state information (values of the attributes), in our map example this might be a legend-specific kilometers per centimeter. Thus, an object is an actual creation of an object class just as the map of Europe circa 1812 is an instance of the class *political map*.

In addition to the basic components, an object will have an interface. The interface is the method with which a user of the object is able to interact with the object. We might view a paper map by placing it on a table and use a ruler to compare distances with the legend. Let us consider how we would analyze the map if it were an object on a computer. The political boundaries might be relatively unchanged, but the entire functionality of the map may have changed dramatically. We might find that to determine distance we need only to mark two points and then request distance. We might aggregate this map with a weather map to project the type of weather that is expected in the near future. Another aspect of the object map is that unknown interests may have access to it and the information that we have extracted from the map. You may not know who labels the map and places or changed information such as the legend. Thus, our traditional concept of maps may be completely challenged. We may lock paper maps in a safe, but not object maps. Therefore, unauthorized access to the map object and its functionality may be a major stumbling block.

Before we examine each of the components of an object, it is important to note one capability of its combination of components. The inclusion of both functionality and attribute values in an object allows the object to maintain the state of the instantiated object at any point in time. The significance of this is that an object which is instantiated by a process can possibly exist independently of that process.

Definition: An object that exits independently from the process that instantiated it is a *persistent object*.

The significance of this feature is that an object might run on a computer, halt, and wait for instruction to activate at a later time. More will be mentioned about this later.

Objects can also be grouped together in order to solve a particular problem. This type of grouping is know as *aggregation* and the group of objects is known as an *aggregate*.

IL2 Object Components and Their Relationship to The Environment

An object will always have four major components:

1. An identity

2. Attributes (and their values)
3. Behavior of the class
4. The interface

Each of these is important in understanding the way objects add functionality to your system and in determining where and how an object has been used in your system.

It is valuable to examine these components to note how each one relates to its software environment and thus may be the source of potentially undesired behavior. The previous map analogy will be utilized in the following paragraphs.

II.2.a Identity

The identity of an object determines the rights and responsibilities of that object. For example, it would not make sense to try to determine the current political boundaries of Czechoslovakia by using a map of Europe circa 1812. The identity is the keystone for binding a set of credentials to an object.

One particular concern when using an identified object is whether it is the correct identity. If a map of Europe circa 1812 is mislabeled 1996, someone might unknowingly try to determine the current boundaries of Germany, Hungary, or Poland using an old map. Thus, mislabeling of an object could lead to an undesired action while using the object.

The identity of an object also allows the object to be stored. This is the critical feature of a persistent object. This allows the storage location or file name to be linked to the particular object entity.

II.2.b Attributes and Values

Attributes and their values are the components that give an object a defined *state*. State is the foundation for decision in a system. An object (and also a program) uses the state of the system to decide which path to take in a decision process. While attributes are static, their values are dynamic. If a value has been affected either wittingly or unwittingly, a decision may be made based on that value.

For example, if the legend of a map is altered, then direction, distance and decisions would be altered to correspond with that state. Changes in the state of an object can have detrimental effects when they alter the state the owner of the object expects.

II.2.c Behavior

Behavior is the set of actions that an object can take. Though the paths may be decided by the state of the system, the total set of possible actions is defined in the behavior set. In fact, the possible behavior is

often the area that can best be examined to determine the expected behavior of an object even though the exact state may not be determined.

It might seem too obvious to point out that in some sense any destructive behavior must be associated with the methods of the object. If destructive behavior exists, it will always be reflected whether in part or in full in the method.

III.2.d The Interface

The interface is the guard to the front door of an object. It determines which behavior or attributes it will expose to external scrutiny or manipulation. In many languages, an object can choose to expose some of its variables to other objects allowing those other objects to inspect and even modify the variables. Also, the interface can be designed to hide methods and attributes from other objects (information hiding), forbidding those objects from invoking certain methods or effecting attribute values.

The process of developing an interface is critical in creating a usable, flexible, and safe object. If the interface is too *open*, it could easily allow other objects to affect data that would change the basic nature of the information contained in it. Conversely if the interface is too *tight*, the object can become less useful and therefore, defeat the purpose of the object. In the map example, if it is decided that only one square kilometer can be examined at a time, a pilot may not have a broad enough scope to make certain decisions. Conversely, if the entire European continent is in view, then there may not be enough detail to make the map useful for routing.

II.3 Object Oriented Programming

For a language to be an object oriented programming language it must allow for the development of objects in an environment that allows *encapsulation, abstraction*, and *inheritance*. Since these are basic properties of object oriented languages, this paper will explain each of these and how they are important in examining object behavior.

II.3.a Encapsulation

Encapsulation is also known as *information hiding*. This is the process by which the client cannot "see" the inside of an object where the implementation has taken place. For example, a map has been displayed and the user does not know whether it has been stored in **jpeg** or **gif** format. The user may not even care what these formats are. The object can maintain private information and methods that can be changed at any time without affecting the other objects that depend on it. This allows a developer to conceal from

external viewing the implementation detail level of an object. Therefore if the developer needs to change from one format to another, the user might never know or care that the change has been made.

II.3.b Abstraction

Abstraction is complementary to encapsulation in that it is the concept of grouping the pieces of a larger entity and then looking at the characteristics of that entity. This allows the users to examine useful characteristics of an object and note how that object might apply to a specific problem the user is trying to solve. This allows the user an exterior view of an object without becoming enmeshed in the details.

Abstraction is an essential approach that allows users to concentrate on what they want to do with the object. However, it also has troublesome aspects for examining behavior. For example, the object may be doing everything you want while also doing a number of tasks you do not want. An internet browser retains a list of the sites you have visited (cookies). You might think this is a convenient reminder of the locations you have visited and allows you to return more easily. While this is true, it also allows web sites to download that information and to determine your interests and thus send you advertisements and mailings. In the map example, a list of the information you extracted could be available to the next user of the map, or someone that enters you network. It might show where you where you "zoomed in" and what information you obtained. Because of encapsulation the user is often unaware of any of these actions.

II.3.c Inheritance

This is the process by which one object class (sub-class) can be formed from another object class (superclass or base class). A sub-class can then give rise to another sub-class and so on. This process, called inheritance, allows a class of objects to obtain methods and attributes, as well rights and responsibilities of a parental class. Inheritance has so many possibly damaging aspects it is discussed again under a later section.

II.4 Destructive Behavior

Destruction in a system can take many forms and be caused in a multitude of ways. While striking the monitor of a computer system with a hammer is destructive, it will not be considered relevant to this work. We are focusing on destruction that is caused by the actions of software within the system, and in particular, software that is in the form of an object. This means that object software actions *undesired* by the *owner* of the system will be considered destructive.

One of the main purposes of security systems is to minimize the amount of destruction that is caused by intruders to a system. Security systems alone do not prevent, nor do they even search for, all destructive

behavior. Their job is to search for *unauthorized* behavior. This is an important point in our definition of destructive objects: unconscious loss of important information is considered just as damaging as a deliberate theft of the same information.

It is important to distinguish the important difference between malevolent and inadvertent destruction. While the end effect may be the same, the intent of the person causing the destruction does play an important role. Malevolent destructive code may seem harder to find than inadvertent destructive code. However, if this is the case, why is it so hard to test code to determine if it meets the specifications of a program? For example, destruction of the type where a programmer multiplies by 3.14 instead of 3.14159., and a rocket doesn't perform properly, may be more difficult to find simply because it is so unintentional and seemingly trivial. Though we will discuss their sources, the solution to these types of errors is almost always the same: better development methods, better walk-throughs, or improved specifications.

Destructive action can come in the following forms: *Data espionage, data corruption, service disruption (denial of service or degradation of service),* and *code corruption.*

II.4.a Data Espionage

Data espionage is the undesired intrusion of a system user to view the data that is "owned" by another user in the system. Database intrusion has been a constant source of espionage because that is where a great deal of information is maintained, but recently with the popularity of networks, real-time information extraction has become a problem. A great deal of data is temporally critical data which, if obtained from a data base, may have little value. However, if it is determined from a volatile source during a time critical period, it may be of great value.

If intruders can enter a data base, they might be able to determine a command that has been issued previously, but, if the intruder can obtain the command in real-time (as it is issued) the value to disrupt or defend is greatly enhanced.

II.4.b Data Corruption

Data corruption is the undesired changing of data in the system. This can be a change in either the real-time change to a variable or parameter, or the changing of a file. If a command in a critical situation can be changed to reflect a different strategy, the possible effect on a battle strategy is enormous.

II.4.c Service Interference

Service interference means that the quality of service is degraded and the performance becomes less valuable to the user as more time elapses. Service interference can be broken down into two well known types of interference, denial of service and degradation of service.

Degradation of service is the process whereby some or possibly all of the usual services that you expect from the computer are still available but they are either slower than expected or the quality (speed, closeness to specification etc.) is not what it should be. This might be seen in the distribution of information taking a greater amount of time than necessary, and therefore not being as valuable if the network had distributed the information in a timely nature.

Denial of service implies that a real-time hard trigger (cut-off time) is either implied or known. If a task cannot be accomplished before the trigger is reached, the value of the task is falls to zero. For example, if the system is preoccupied performing other tasks, the user is denied the needed service. This is usually accomplished by getting the host system to run other non-interruptible services. If an intrusion to a system with sensitive information is detected, it may be important to stop the loss of data. If the intruder is not stopped soon enough, the value of stopping that person may be lost.

II.4.d Code Corruption

Code corruption is the act of changing the code of a procedure or function in a manner that is undesired by the owner of the software. This means that the software that is expected to deliver a particular task may not execute that task, but may execute a different task. Code corruption can also simply crash the program.

II.5 Destructive Objects

Now that we know what a destructive act is, we can define objects that cause these acts.

Definition: A *destructive object* is an object or aggregate that causes a destructive action in the system.

While this definition leaves open the differentiation between a malevolent and a inadvertent destructive object, it does allow the inclusion of objects that cost money, time, or the loss of sensitive information even if the intent of the user is honorable.

An example of a destructive object is an object that is supposed to fill a simple triangle with a shade of blue but instead it fills the triangle with a shade of red. This might appear to be a simple mistake, and probably easily corrected, but if it happens on a map of a conflict area it may result in an attack upon friendly forces. Another destructive object might be an applet (a Java™ program part) that captures information from your computer as you browse the web.

III. Creating Destructive Objects

When a user creates an object there is always the potential to create a destructive object. We will examine how objects are created and grouped to examine where destructive objects come from. There are three ways to develop an object or object group:

1) write a stand alone object class (instantiation)
2) create an object from a set of other objects (an aggregate)
3) inherit properties from another object class (inheritance)

Since these are basic approaches to using classes in a system, let us examine how destructive properties could be brought into a system from each one of these structures.

III.1 Destruction Through Instantiation

Individual objects are created (instantiated) from an object class. This is the process of creating an actual object from an abstract concept (class). This object has a real identity and a actual set of credentials. This is similar to creating an actual integer (say 7) from the predefined type integer. An individual object can contain both methods and attributes obtained from the object class as well as the values for some of the attributes.

An object, therefore, contains the code from the object class. A likely source of destructive behavior in an object is the generation of bad code. This is often not the intent of the programmer, but is the result of poor design, coding, or testing. This type of destructive behavior can be as damaging as any in the system, as all too often the assumption on the part of developers is that code written and compiled is code that is tested and meets specifications. In fact, much of the destructive behavior of objects, though not malevolent, can be attributed to this. The solution is as mentioned, a better process in developing the software.

Through the creation process, an object can obtain any known virus from its parental class. What makes this different from a traditional virus is that the object virus can reside within one, two, or many methods thus making it more difficult to detect with the usual virus scan software.

Another area of concern in an object is the possibility of self-modifying code. Self-modifying code is code that is not harmful in and of itself, but has the potential for great destruction as it allows code a metamorphosis after it has passed through language or system checks. This can be different from a virus in that self-modifying code is often a valid approach to minimize space or execution time. One could argue that simply is code that should not be allowed, and code that can be checked for quite easily, it is possible and sometimes necessary.

III.2 Destruction Through Aggregation

The process of grouping two or more objects together allows a user to "build" the software solution to a problem. The grouping process adds several interesting questions that could point to destructive behavior. How well do the individual objects work? How well do the objects work as an aggregate? How much information does each one need to perform the required duties? How well are the interfaces written and how much complexity do they add? What is the genesis of each object, and can the actions of the object be trusted? Can the actions of the aggregate be trusted?

Here encapsulation plays an important role in the destructive scenario. If a user chooses to execute an object, to accomplish a desired task, other behaviors may also be executed. Destructive behavior may be introduced not from the action desired, but from the hidden actions of the object; the encapsulated features unknown to the user.

Destructive properties may also be introduced through the interface of the objects. If the interface from one object to another is not correct, the system may fail either because the correct data was not passed, or the data was misinterpreted when it arrived at the receiving object. This is particularly common when executable objects are linked together. The run time linking of objects is a practice that allows a maximal flexibility in the code, but adds this potentially dangerous interface problem.

An additional problem arises when several objects work together. When each individual object performs its specified task, the combination of objects may conflict. This is sometimes seen when software needs to use a port, but another piece of software is running and has already grabbed that port and will not release it until the program ends. An analogy to this might be the taking of drugs to assist in your health. Taking one drug might help an ailment, but taking several drugs, each with its own task, might do immeasurable damage to your health.

One problem that can take place in an aggregate is that an object passes another object a message in an effort to perform a task. While this is not destructive, it is dangerous. It could allow the first object to perform the task, while not taking responsibility for that task. This, in turn, would allow the user to repudiate any responsibility.

Another possible problem that is related to repudiation is spoofing. Spoofing is the act of tricking the system or another object into either believing the object either has different credentials (more rights) than it does or that its identity is different that what it is. An example might be if an object wishes to have a server execute and action the object is not allowed to perform. The object might request a second object that does has the capability to request that from the server, and pass the results to the first object. The server might allow the action to be performed because it based its decision on the credentials of the second

object. In this way the first object spoofed a server into performing a task that it was not allowed to perform.

A major source of problems in objects is the interface problem. This comes in two major types: *mode* and *type*. The *mode* problem involves messages being passed between objects where the object receiving the message needs to examine the data based on the state of the object. The object might have two states, one for regular tasks and one to handle exceptions such as divide by zero. If an exception is determined, the object would not send the normal messages, but one that says the "I found an error". If the receiving object does not realize it is in a correction mode, it might try to consider the error message as regular data. If an object needs to know information about timing or it is possible to miss a signal and perform destructive actions based on that.

Another problem occurs if the *types* of parameters do not match. In passing parameters, if a receiving parameter is expecting an integer and receives a real number, there can be a problem in the interpretation of the data. In addition to the problem of simple type matching, the use of pointers as parameters adds much more complexity. Under the heading of passing parameters is the practice of passing a pointer to data, a function, or an object. This may appear to be no more of a problem than the others mentioned, but this provides a path to add almost any feature to an object at run time.

There are several possible scenarios that may add destructive behavior using the aggregate model. One possibility is that of destructive behavior contained in several objects. Each component of this behavior is unintentional or appears inadvertent, but together, the components supply the pieces needed for destructive behavior.

In a dynamic programming environment, objects are created, linked, and utilized at run-time. Making the selection as to which object to link is a dynamic decision, therefore, testing the program becomes as difficult as testing all possibilities. To add to that difficulty the search for a named object is usually done in a "closest" first approach. One method is to check in the file of the object that wished to form the link, then the next "closest," et cetera. This approach seems sensible, but, because it is done at run-time, the programmer may not know if the objects are linking as they expected.

For example, assume an object **foo** wishes to link to object **greenwich,** located somewhere in England. This is important as **foo** wants the absolutely correct time for a command to be generated. This would normally be no problem, except that a local persistent object **greenwich** has been generated for testing purposes, and placed in the same file as **foo**. **Foo** will be linked to the closest **greenwich** object. Therefore, the object **foo** will be to local time, and the correct time may never be linked to **foo**.

Another possibility is that since aggregates can be formed statically or dynamically, a set of processes may have been run before without the destructive component being added to the system. Once the critical component is added, the behavior can take place. This component might only be a catalyst to the behavior. The possibility exists that the new or updated component could add the missing ingredient to a lethal daisy chain virus.

III.3 Destruction Through Inheritance

Inheritance is one of the most useful aspects of object oriented programming. It allows the creator of an object to reuse the characteristics of one class (the base class) and create another class (sub-class) with some or all of those characteristics. The creator does not need to rewrite the code or add attributes to obtain the full functionality of the base class. Therefore, given the privileges, the sub-class has complete use of the superclass and may add or subtract functionality and attributes as desired. The sub-class can also be denied privileges to the superclass' functions and attributes. For example a data base management object might be created that restricts access to certain sensitive data while otherwise having the same characteristics of the original database management system.

The creation of an object class by inheriting properties from one or more superclasses allows the possibility of destructive properties in several interesting ways:

1) If the sub-class can override the constraints placed on it from the superclass, it now can have the superclass' functionality
2) The sub-class may lose the actual functionality when certain privileges are taken away, thereby becoming ineffective at its task
3) Through the act of multiple inheritance the sub-class may gain or lose functionality because of the interaction of a incorrect set of inherited methods
4) The sub-class could be a viral class due to the interaction of the new methods
5) The sub-class could try to repudiate responsibility and attribute it to the parent (superclass)

We will examine these scenarios individually to see how a destructive process might develop from each.

1) A common practice is to develop an object class that has a great deal of functionality and then not allow inheriting classes to have that complete functionality. This allows a developer to maximize the highest level for functionality and then simply limit the sub-classes to the specific functionality needed to do their job. One language limits the disk access for sub-classes of a certain type. This effectively assures that if a programmer develops an object from the original, the sub-class will not be able to have the superclasses full capability. If, however, the sub-class can make the system think it is

the superclass, it will have access to the files. Moreover, it can access files when the users of that object think it cannot access files.

2) While crippling the sub-classes can work in allowing the development of certain objects, it can have the effect of lowering the utility those same classes. If files cannot be touched by a class of objects, than it is clear that the utility of those objects is very limited. It may be that the sub-classes have a requirement for temporary file access or the software is much less effective without that access. Therefore, limiting the credentials for subclasses can have destructive effects on the overall utility of the objects.

3) Multiple inheritance brings a special set of problems with it, as well as a good deal of flexibility. Two types of problems are critical to this type of inheritance: resolving attributes or methods, and method interaction. Assume a sub-class **foobar** has inherited all the methods from both **foo1** and **foo2**. If **foo1** and **foo2** both have a method called **method1** then it is not clear which **method1** (**foo1**.method1 or **foo2**.method1) **foobar** has inherited. This means the **foobar** may act differently depending on the implementation of the language.

4) In single and multiple inheritance, one problem that arises is the interaction of methods. In single or multiple inheritance, methods can be added to the sub-class that has been created. This poses the problem that the new methods may conflict with the methods of one or both parents. If a certain method maintains a repository for information and it uses a different method to assure the validity of the information, a new method created could easily conflict with the method maintaining the validity. This can happen as a new method, or if a class has added a new capability by inheriting the new method.

5) A newly created sub-class may try to exercise functionality that it is not allowed to have, then repudiate responsibility. An important aspect of security is recovery, and thus involves finding who is responsible for what action. If an object can "blame" the action on a parental class, it can repudiate the action, thus avoiding responsibility.

IV. Approaches to Avoiding Destructive Behavior

Two newer methods of using objects, JAVA ™ and CORBA ™ are considered here for their respective approaches to considering destructive behavior. While each of these approaches has capability beyond what is mentioned we considered these as pure approaches to the problem of destructive bahavior.

In a system that considers destructive behavior, there are three stages of dealing with the undesired action: avoidance, detection, and recovery. Avoidance is the process of not allowing the software to contain

destructive actions. This would normally be done in the design, coding (and compilation), linking, and testing phases. Detection is the process of dealing with code that has destructive actions internal to the code. This could include avoiding the affects of destructive behavior that are contained in the code. Recovery includes undoing damage as well as determining responsibility for whatever actions were taken, and determining strategies for the future.

We have examined two major approaches of object use and noted the major techniques of avoiding destructive behavior. These approaches can be broadly categorized as *language based* and *language independent*. In both cases the process of avoiding destructive behavior in the objects is not analyzed as they are both ways of using objects, and not methods of developing objects.

The language based approach places the process of detection of destructive behavior on the constructs of the language, or the environment under which the language is interpreted or is executed. The language independent approach places the emphasis on an object system interface and the architecture of the system.

IV.1 The Language Dependent Approach

The language dependent approach that was studied uses the *write once run anywhere* model for distributed computing. In this model the user runs objects on a local machine. The objects may be developed in a remote location, requested, and brought to the users machine for execution. (The approach we examined has a method to run objects remotely, but this will not be considered here.) Running an object locally can add a great deal of speed as it is not encumbered by traffic on any network. Running an object locally means that the code for the object must be brought to the users machine. Using the *write once run anywhere* technique, the object is translated into a standard (intermediate) code before it is allowed to run on the users machine. The standard code is not optimized for a particular machine or operating system and therefore is slower than code compiled as native to a particular system.

The technique of translating a programming language to a standard intermediate language allows the user to check the code as it enters their machine. The code can then be passed through the security management levels (filters) to be checked for undesirable properties. This can eliminate several types of destructive behaviors: it can check for unauthorized file access, self modifying code, and calls to disallowed objects.

While this is an excellent approach, it also means having the objects on your machine. If the security manager does not check for specific destructive properties, they may well be present in the code.

This approach does have some methods that lend themselves to the recovery process. In particular it is usually possible to determine where the identity of the offending object and thus track the guilty object to the source.

IV.2 Language Independent

The language independent approach examined uses a *middleware* object approach, where a software service works between an object that requests an action and another object executing the action. We will refer to the requesting object as local and the object executing the request as remote. (While the concept of local and remote are not necessary to the middleware, it is in the spirit of the approach and allows a better explanation.) The remote objects is run on the machine that it originally resides on, and the data requested is passed across the middleware according to an (IDL) Interface Definition Language to the local object.

This approach solves a great many concerns about destructive objects as the objects do not run on the users system. If the offending object formats its remote drive or binds up its remote system, that would minimize the effect on the users local system. The only place for destructive data to enter into the users system is through the IDL. This requires diligence on defining what allowable data can enter through the IDL, but this greatly reduces the possible set of destructive actions an object can take. Does this eliminate all other destructive actions? No.

In this type of system a designer cannot fully define any software aggregate as the remote objects are run on remote systems by remote operating systems. Consider for example, if the timing of an objects response is critical to the software solution, the designer must now control the users machine and the remote machine and the traffic on the network.. The current state of this approach does not allow this type of control, and it is would be difficult at best. Since the user cannot define these types of problems it also implies the inability to handle reliability as well as recovery of a crash in the software.

V. Conclusions

Object oriented technology has produced a great deal of optimism in the software community due to the promise of reusable quality code generated from off the shelf components. These components will lead to a more engineering approach to programming and software design. This is turn will lead to a time of productivity not known before in the field. While this may well be true, the use of objects has risks that are not well known and not well considered. This report is a start to examining and understanding where destructive properties may arise in our use of this new technology.

Destructive objects can arise from almost any phase in the development of an object. Each new innovation and technique has potential for destructive behavior that must be considered in any risk analysis when using objects and in the process of designing software that uses objects.

It would be valuable to examine the best approaches in each case to minimize the potential for destructive behavior. In some cases the use of pre-execution software can be a valuable tool as was discussed in the language dependent approach to object oriented development. This has yet to be shown as a complete solution. The use of external objects linked across a network is certainly another valuable approach. Some problems with this approach need to be examined for solutions. Would a hybrid model work and under what circumstances would that be most viable?

VI. Acknowledgments

I would like to thank my colleagues at the Information Institute at Rome Laboratory for their advice and assistance. In particular, I would like to thank Heather Dussault for allowing me to take part in this project, for all of her assistance in developing the direction for this work, and for her corrections and suggestions in the work. My appreciation also goes to Joe Giordano for his subsequent directorship and insightful suggestions.

References

[Adl95] R. M. Adler, "Distributed Coordination Models for Client/Server Computing", *Computer,* pp. 14 - 22, April, 1995.

[Bu97] T. Budd, *An Introduction to Object-Oriented Programming* 2nd ed., Addison Wesley 1997

[CS97] L. T. Chen and T. Suda, "Designing Mobile Computing Systems Using Distributed Objects", *IEEE Communications Magazine,* pp. 62 - 70, February, 1997.

[FBDW97] E. Felten, D. Balfanz, D. Dean, and D. S. Wallach, (23 July, 1997), Department of Computer Science, Princeton University, "Web Spoofing: An Internet Con Game" [WWW Document], URL *http://www.cs.princeton.edu/sip/spoofing.html*

[Fla97] D. Flanagan, *Java In A Nutshell,* Cambridge: O'Reilly, 1997.

[Gra97] M. Grand, *Java Language Reference,* Cambridge: O'Reilly, 1997.

[Lew95] T. G. Lewis, "Where Is Client/Server Software Headed?", *Computer,* pp. 49 - 55, April, 1995.

[MF97a] G. McGraw and E. W. Felten, *Java Security,* New York: Wiley, 1997.

[MF97b] G. McGraw and E. W. Felten, "Plugs for Java's Security Holes", *Byte,* Jan., 1997.

[MF97c] G. McGraw and E. W. Felten, "Java Security and Type Safety", *Byte,* Jan., 1997.

[MF97d] G. McGraw and E. W. Felten, "Understanding the keys to Java security -- the sandbox and authentication", *JavaWorld,* May, 1997.

[MS97] S. Maffeis and D. C. Schmidt, "Constructing Reliable Distributed Communication Systems with CORBA", *IEEE Communications Magazine",* pp. 56 - 60, February, 1997.

[MZ94] T. J. Mowbray and R. Zahavi, *The Essential CORBA,* New York: Wiley, 1994.

[Rei96] Arthur J. Riel, Object-Oriented Design Heuristics, Addison-Wesley Publishing, 1996

[Ven97] B. Venners, "Java's Security Architecture", *JavaWorld,* August, 1997.

[Zha97] X. N. Zhang, "Secure Code Distribution", *Computer,* pp. 76-79, June, 1997.

SIMULATION OF A ROBUST LOCALLY OPTIMUM RECEIVER IN CORRELATED NOISE USING AUTOREGRESSIVE MODELING

Donald R. Ucci
Associate Professor
Department of Electrical and Computer Science Engineering

Illinois Institute of Technology
3300 S. Federal Street
Chicago, IL 60616-3793

Final Report for:
Summer Research Program
Rome Laboratory

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, Washington, DC

And

Rome Laboratory

September 1997

SIMULATION OF A ROBUST LOCALLY OPTIMUM RECEIVER IN CORRELATED NOISE USING AUTOREGRESSIVE MODELING

Donald R Ucci
Associate Professor
Electrical and Computer Engineering Department
Illinois Institute of Technology

Abstract

This report presents the simulation of a robust *locally optimum* (LO) non-linear spread spectrum receiver. The signaling environment consists of the desired received signal in correlated interference and thermal noise. Autoregressive (AR) spectral modeling methods and a histogram approximation of the probability density function are employed. Preliminary results for transmission in the presence of *continuous wave* (CW) and *partial band* (PB) interference are presented and discussed. A comparison of this method to a similar nonlinear processing algorithm is performed. Preliminary results for the performance of a binary phase-shift keyed communications system indicate that applying AR modeling to the environment improves performance substantially, especially in the case of partial band interference.

SIMULATION OF A ROBUST LOCALLY OPTIMUM RECEIVER IN CORRELATED NOISE USING AUTOREGRESSIVE MODELING

Donald R. Ucci

The AR LO Detector

Introduction

Many facets are involved in the design of a spread spectrum communications system [20]. One important consideration is determining a method to best recover the transmitted signal when it is subjected to interference in the transmission path. This interference is often highly correlated and not necessarily Gaussian as in typical interference models [12]. The linear correlator realization of the matched filter is no longer optimal for this environment. *Locally optimum* (LO) detection provides a method for circumventing this problem if the *probability density function* (pdf) of the interference is known. However, in cases where the interference exhibits strong self-correlation, traditional LO methods exhibit poor performance [24].

LO detectors with memory more successfully combat this type of disturbance. The disadvantage of a memory-based processor is the rapid increase in the dimensionality of the *joint probability density function* (jpdf) noise vector making the LO detector nonlinearity unwieldy [22]. To alleviate this problem, frequency domain methods are used to determine a P^{th} *-order autoregressive* (AR(P)) model of the channel disturbance. The AR methodology reduces

the dimensionality of the underlying jpdf to that of the model order. The statistics of the jpdf remain an issue. To this end, pdf estimation techniques for *independent identically distributed* (iid) noise samples prove useful (11) since the input sequence to the AR model is iid, albeit characterized by some unknown pdf. The necessary AR model parameters are determined by well-known spectral estimation techniques [12]. The following sections describe the basic LO detector, develop the *autoregressive locally optimum* (ARLO) detector, discuss the simulation results, and present the conclusion.

The Locally Optimum Detector

In communicating real, binary data in additive noise, the receiver must distinguish between two possible information signals. For example, in a *binary phase shift keyed* (BPSK) communications system, the receiver must decide whether a value of ± 1 was transmitted.

More formally, the detector must choose correctly between one of the following hypotheses:

$$\begin{aligned} H_1&\text{: Signal } \mathbf{s}_1 \text{ is present} \\ H_0&\text{: Signal } \mathbf{s}_0 \text{ is present,} \end{aligned} \tag{1.1}$$

where the notation $\mathbf{x} = [x_1 \cdots x_N]^T$ denotes a vector of length N and $\mathbf{s}_1$ and $\mathbf{s}_0$ represent the two possible signal vectors. The value of N, in general, denotes the number of signal samples for a given symbol (for BPSK, a symbol is represented by a bit). To derive the corresponding LO detector with memory, begin by observing the received signal vector as:

$$\mathbf{r} = \mathbf{s}_m + \mathbf{n}, \tag{1.2}$$

where $m = 0$ or 1 and $\mathbf{n}$ is a random noise vector, with jpdf, $f_{\mathbf{n}}(\boldsymbol{\eta})$. Given the observation, $\mathbf{r} = \boldsymbol{\rho}$, the optimum detector is of the form,

$$L(\boldsymbol{\rho}) \underset{\text{choose } H_0}{\overset{\text{choose } H_1}{\gtrless}} \tilde{\gamma}. \tag{1.3}$$

where $\tilde{\gamma}$ is the decision threshold and is zero for *maximum likelihood* (ML) detection. In Eq. (1.3), $L(\rho)$ is the log-likelihood ratio, given by,

$$L(\boldsymbol{\rho}) = \ln\left[\frac{f_{\mathbf{r}}(\boldsymbol{\rho} \mid H_1)}{f_{\mathbf{r}}(\boldsymbol{\rho} \mid H_0)}\right] = \ln\left[\frac{f_{\mathbf{n}}(\boldsymbol{\rho} - \mathbf{s}_1)}{f_{\mathbf{n}}(\boldsymbol{\rho} - \mathbf{s}_0)}\right], \tag{1.4}$$

where $f_{\mathbf{r}|\mathbf{H}}$ is the jpdf of the received signal under hypothesis H. Using a first-order Taylor series expansion around the signal points, and assuming a large *interference-to-signal* (ISR), the log-likelihood ratio can be approximated by $L(\rho) \approx l(\rho)$ [12] where,

$$l(\boldsymbol{\rho}) = \sum_{i=1}^{N} (s_{1i} - s_{0i}) g_i(\boldsymbol{\rho}). \tag{1.5}$$

The s_{ji} in Eq. (1.5) represents the i^{th} sample of the j^{th} signal and

$$g_i(\boldsymbol{\rho}) \triangleq -\frac{\partial}{\partial \rho_i} \ln[f_{\mathbf{n}}(\boldsymbol{\rho})] \equiv \frac{-\frac{\partial}{\partial \rho_i} f_{\mathbf{n}}(\boldsymbol{\rho})}{f_{\mathbf{n}}(\boldsymbol{\rho})}, \tag{1.6}$$

s the LO nonlinearity. Thus, the LO detector for the detection of known binary signals in additive noise is,

$$l(\boldsymbol{\rho}) = \textstyle\sum_{i=1}^{N} (s_{1i} - s_{0i}) g_i(\boldsymbol{\rho}) \underset{\text{choose } H_0}{\overset{\text{choose } H_1}{\gtrless}} \tilde{\gamma}. \tag{1.7}$$

The LO Detector for AR Noise of Order P

Suppose the noise sample, n_i, at a discrete time instant, i, can be modeled as a P^{th}-order Markov process. Then the autoregressive noise model is [24]

$$n_i = \begin{cases} \sum_{j=1}^{P} a_j n_{i-j} + w_i, & \text{for } i \in [1, N], \\ 0, & \text{for } i \leq 0 \end{cases} \tag{1.8}$$

where the $\{a_i\}$ are the AR(P) coefficients, w_i represents an iid random process, and the vectors contain N samples. In this case, the noise pdf is,

$$f_{\mathbf{n}}(\boldsymbol{\eta}) = f_{\mathbf{n}}(\eta_1, \cdots, \eta_N) = \prod_{i=1}^{N} f_{n_i}(\eta_i \mid \eta_{i-1}, \cdots, \eta_{i-P}), \tag{1.9}$$

where,

$$\begin{aligned} f_{n_i}(\eta_i \mid \eta_{i-1}, \cdots, \eta_{i-P}) &\triangleq f_{n_i}(\eta_i) && \text{for } i = 1, \\ f_{n_i}(\eta_i \mid \eta_{i-1}, \cdots, \eta_{i-P}) &\triangleq f_{n_i}(\eta_i \mid \eta_{i-1}, \cdots, \eta_1) && \text{for } i = 2, \cdots, P. \end{aligned} \tag{1.10}$$

If a "block approximation" [12] [1] is assumed, then $f_{\mathbf{n}}(\boldsymbol{\eta})$ can be written as,

$$f_{\mathbf{n}}(\boldsymbol{\eta}) = \prod_{i=1}^{N} f_w \left(- \sum_{j=0}^{P} a_j \eta_{i-j} \right), \tag{1.11}$$

where $f_w(\omega)$ is the pdf of the white noise process and $a_0 \triangleq -1$. For this case, the form of the nonlinearity is,

$$g_i(\boldsymbol{\nu}) \triangleq \sum_{l=0}^{\min(P, N-i)} a_l h' \left(- \sum_{j=0}^{P} a_j \nu_{i+l-j} \right) \text{ for } i \in [1, N], \tag{1.12}$$

where,

[1] In general, η_1 depends on the last "P" previous samples, which are assumed to be zero so that, $\eta_1 = w_1$. This only affects a small fraction of the terms for $N \gg P$.

$$\nu_k = \begin{cases} 0, & \text{for } k \leq 0, \\ \rho_k & \text{for } k \in [\,1,\, N\,], \end{cases} \tag{1.13}$$

$h'(\omega)$ is the derivative of $h(\omega)$, and

$$h(\omega) \triangleq \ln\left[f_w(\omega)\right]. \tag{1.14}$$

Histogram and AR Estimation Techniques

Essential to the computation of the LO detector nonlinearity is the noise pdf, $f_w(\omega)$, and the autoregressive coefficients, $\{a_i\}$. Since neither the source statistics of the driving white noise, nor the AR coefficients are known *a priori*, they must be estimated.

The estimation of the pdf is performed via a histogram method employed in the LO detector without memory [12]. Pure noise data is not available, so the received data must be used. Then, a three-point derivative is used to to compute $h'(\rho)$. The histogram approach is chosen because of its ease and simplicity of implementation and generally acceptable performance.

The AR coefficients are estimated using the *modified covariance algorithm* (MCA) [5]. This is a non-windowing method similar to the covariance method, but it differs from the latter in that it minimizes the sum of the squares of the forward and backward predictor errors. The MCA is chosen since, when compared to other AR estimation methods, it often provides a) stable, high resolution spectral estimates; b) exhibits lower sensitivity to phase and decreased peak shifting, and, c) is not subject to spectral line splitting.

Simulation Results

Simulation of an ARLO detector in a spread spectrum communications system comprised of a transmitter, channel, and (assumed) synchronized receiver was performed. In this system, the transmitter emits B BPSK symbols sampled at N_b samples per symbol. This sample waveform is then multiplied by a *pseudo-noise* (PN) sequencer with a chipping rate of N_c samples per chip. Thus, the SS system processing gain is $PG = N_b/N_c$. A scale parameter for the corresponding *Signal-To-Noise Ratio* (SNR) is used to generate probability of error curves.

The channel is assumed to be an *Additive White Gaussian Noise* (AWGN) channel that is additionally corrupted by a self-correlated interfering signal which is either a *continuous wave* (CW) interferer or a *partial band* (PB) interferer. The parameter for the CW interferer is its frequency whereas for the PB interferer the passband is specified. Both interferers are scaled to a pre-determined ISR. The receiver is assumed to be synchronized with ideal filtering at the front end. The received signal samples are then passed to the ARLO detector.

The ARLO detector estimates the statistics of the received signal using a histogram method [9] to obtain an approximation of the pdf. A three-point derivative of the likelihood function, taking advantage of the properties of natural logarithms, is made. Next, an estimate of the AR model parameters is obtained using the modified covariance algorithm [5] with P chosen as a parameter. The received signal is then filtered using the $\{a_i\}$ as a *finite impulse response* (FIR) filter to whiten the spectrum. Then, Eq. (1.12) is applied to this whitened signal to determine the LO nonlinearity. From this, a decision is made using a standard PN decoder.

In the CW case, the interference is modeled as a sinusoid at a given frequency within the main lobe of the signal spectrum. Typical spectra for

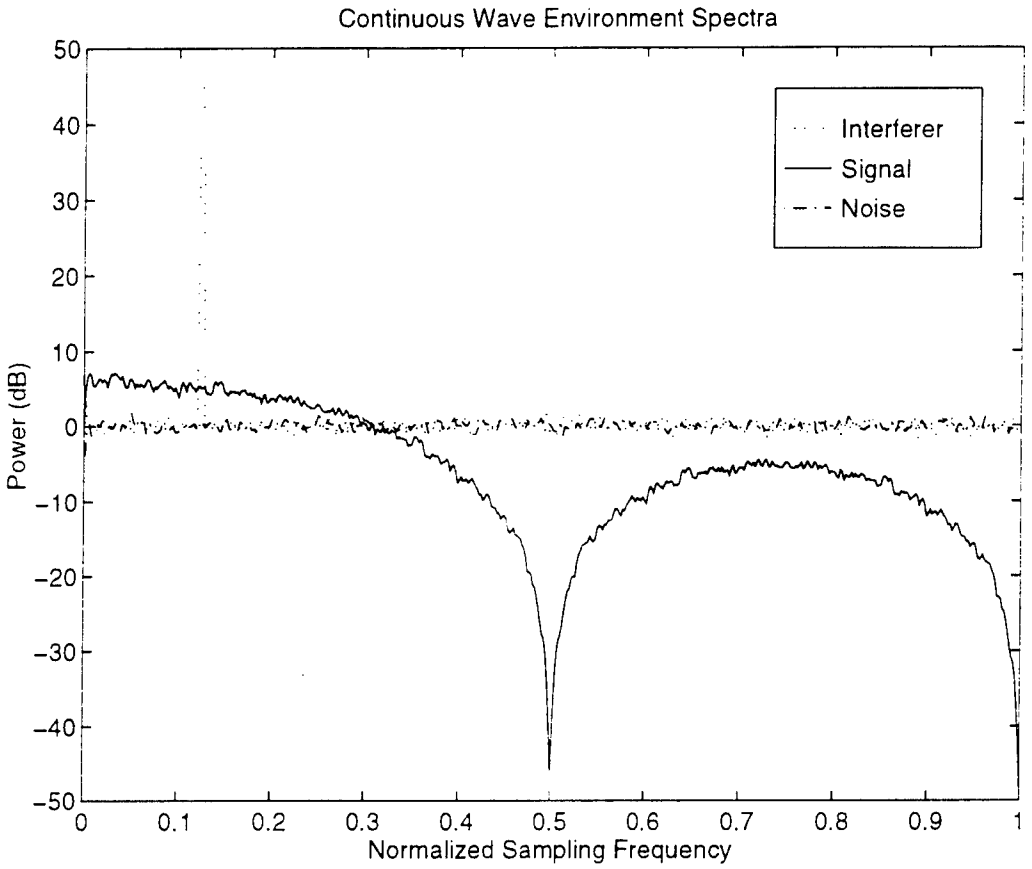


Figure 1.1: CW environment for the case of ISR = 20dB with PG = 12dB

this environment are depicted in Figure 1.1.

A typical bit error performance curve for this environment is shown in Figure 1.2, for $P = 16$. Bit error curves are provided for a standard linear correlator, the LO detector without memory, a linear correlator prefixed by a whitening filter, an LO detector prefixed by the same whitener and the ARLO detector. The whitening filters are created from the same AR coefficient estimates used in the ARLO detector.

In the PB case, the interference is modeled as an iid random process passing through an all-pole filter of pre-determined coefficients. Figure 1.3 depicts typical spectra in the CW environment.

A typical bit error performance curve for the PB environment is shown

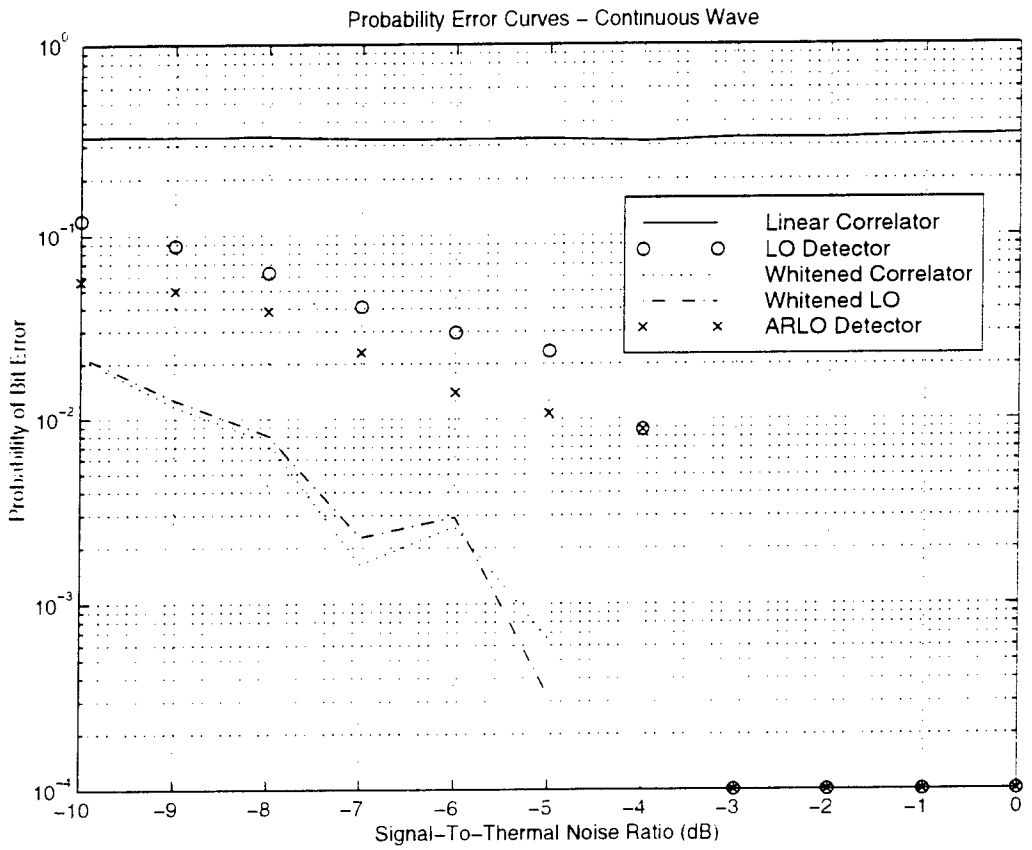


Figure 1.2: CW environment for the case of ISR = 20dB with PG = 12dB

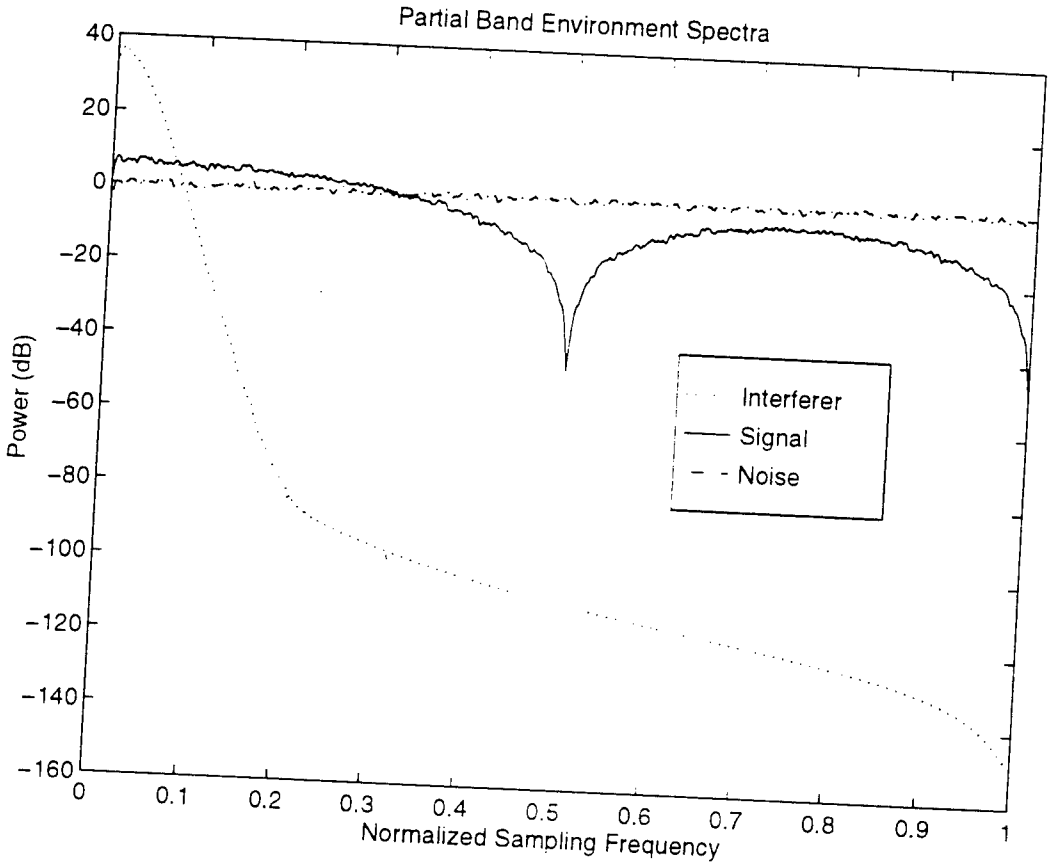


Figure 1.3: PB environment For the case of ISR = 20dB with PG = 12dB

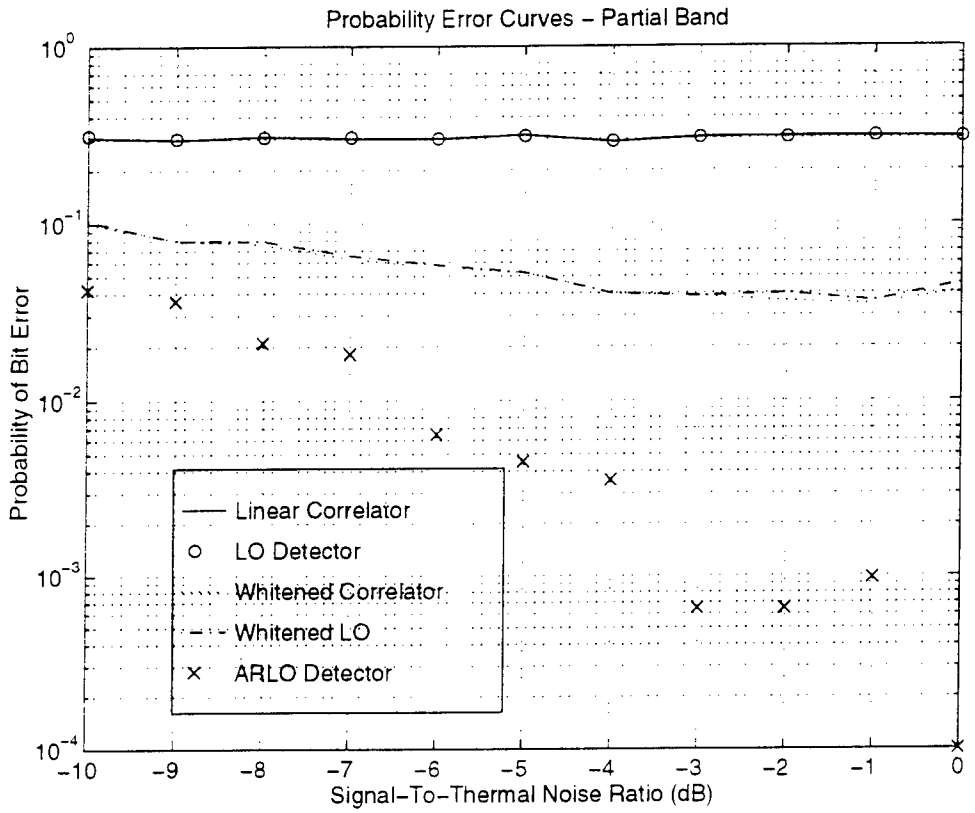


Figure 1.4: PB environment for the case of ISR = 20dB with PG = 12dB

in Figure 1.4. The parameters for this particular observation are $P = 16$ AR coefficients for the model and a pass band roughly equivalent to half of the main lobe of the signal spectrum. The performance for the same five detectors studied in the CW environment are illustrated.

Conclusions

Analyzing the bit error curves in the previous figures for CW interference, we see that auto-regressive techniques do not seem to provide an advantage over traditional LO detection methods. In fact, it appears that whitening filters provide slightly better performance. However, the difference in performance is not significant. It is obvious that the linear correlator performs much more

poorly than any of the other techniques. These results match those obtained in [12] for similar environments.

However, in studying the cases for PB interferers, we note a dramatic increase in performance of the ARLO detector. For a given value of bit error probablity, the ARLO detector requires a significantly lower SNR than other LO detection schemes. Even the whitening methods cannot excise the wider-band interferer. Thus, the ARLO detector is more robust than any of the other receviers studied.

Bibliography

[1] P.M. Clarkson, *Optimal and Adaptive Signal Processing*, CRC Press, Inc., 1993.

[2] P.M. Clarkson abd H. Stark, eds., *Signal Processing Methods for Audio, Images and Telecommunications*, London: Academic Press, 1995.

[3] J.F. Doherty and H. Stark, "Direct-Sequence Spread Spectrum Narrowband Interference Rejection Using Property Restoration," *IEEE Transactions on Communications*, vol. COM-44, pp. 1197-1204, Sep. 1986.

[4] J.H. Grimm, et. al., "Continuous Polynomial Approximation," *Proceedings of the 1993 IEEE MILCOM Conference*, pp. 283-287, 1993.

[5] M.H. Hayes, *Statistical Digital Signal Processing and Modeling*, New York: John Wiley and Sons, 1996.

[6] Hazeltine Report No. 6662, *Adaptive Nonlinear Coherent Processor Development*, Final Technical Report for Rome Air Development Center, USAF Report No. RADC-TR-89-387, 1990.

[7] J.H. Higbie, "Adaptive Nonlinear Suppression of Interference," *Proceedings of the 1988 MILCOM Conference*, pp. 23.3.1-9, 1988.

[8] D.M. Hummels and J. Ying, "Locally Optimum Detection of Unknown in Non-Gaussian Markov Noise," *Proceedings of the IEEE Midwest Symposium on Circuits and Systems*, pp. 1098-1101, 1991.

[9] W.E. Jacklin, J.H. Grimm, and D.R. Ucci, "The Simulation of a Two-Dimensional Spread Spectrum System With Locally Optimum Processing," *Proceedings of the 1993 IEEE MILCOM Conference*, pp. 288-292, 1993.

[10] W.E. Jacklin and D.R. Ucci, "The Fourier Series Implementation of a Locally Optimum Detector," *Proceedings of the 1994 IEEE MILCOM Conference*, pp. 992-996, 1994.

[11] W.E. Jacklin and D.R. Ucci, "A Comparison of Performance Metrics for Two Robust Locally Optimum Detectors," *Proceedings of the 1995 IEEE MILCOM Conference*, pp. 165-169, 1995.

[12] W.E. Jacklin, "Statistical Methods for Robust Locally Optimum Signal Detection," Ph. D. Dissertation , The Illinois Institute of Technology, Jul. 1996.

[13] S.A. Kassam, *Signal Detection in Non-Gaussian Noise*, New York: Springer-Verlag, 1988.

[14] J. Ketchum and J. Proakis, "Adaptive Algorithms for Estimating and Suppressing Narrowband Interference in PN Spread Spectrum Systems," *IEEE Transactions on Communications*, vol. COM-30 , pp. 913-923, May 1982.

[15] A.W. Maras, "Locally Optimum Bayes Detection in Ergodic Markov Noise," *IEEE Transactions on Information Theory*, vol. 40, pp. 41-45, Jan. 1994.

[16] J.L Melsa and D.L. Cohn. *Decision and Estimation Theory*, McGraw-Hill, Inc. 1978.

[17] D. Middleton, "Canonically Optimum Threshold Detection," *IEEE Transactions on Information Theory*, vol IT-12, pp. 230-243, Apr. 1966.

[18] L. Milstein and R. Iltis, "Signal Processing for Interference Rejection in Spread Spectrum Communications," *IEEE Acoustics, Speech and Signal Processing Magazine*, vol. 3, pp. 18-31, Apr. 1986.

[19] H.V. Poor, *An Introduction to Signal Detection and Estimation*, 2^{nd} Ed., New York: Springer-Verlag, 1994.

[20] D. Schilling et. al., "Spread Spectrum for Commercial Communications," *IEEE Communication Society Magazine*, pp. 66-79, Apr. 1991.

[21] D.R. Ucci, W.E. Jacklin, and J.H. Grimm, *Investigation and Simulation of a Nonlinear Processor for Spread Spectrum Receivers* , Final Technical Report for Laboratory, USAF Report No. RL-TR-93-258, 1993.

[22] D.R. Ucci, W.E. Jacklin, J.H. Grimm, *A Spread Spectrum Receiver With Nonlinear Processing*, Final Technical Report for Rome Laboratory, USAF Report No. RL-TR-93-50, 1993.

[23] D.R. Ucci, W.E. Jacklin, and J. Tamas, *Quasi-Optimal Processing in Spread Spectrum Environments* , Final Technical Report for Rome Laboratory, USAF RL Contract No. F30602-93-C-0099, 1994.

[24] D.R. Ucci and W.E. Jacklin *Robust Locally Optimum Detection*, Final Technical Report for Rome Laboratory, USAF RL Contract No. F30602-93-C-0099, 1995.

A PROCESS ENGINEERING APPROACH TO CONTINUOUS COMMAND AND CONTROL ON SECURITY-AWARE COMPUTER NETWORKS

Nong Ye
Assistant Professor
Department of Mechanical Engineering

University of Illinois at Chicago
842 West Taylor Street
Chicago, IL 60607

Final Report for:
Summer Faculty Research Program
Rome Laboratory

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Rome Laboratory - Information Institute

August 1997

A PROCESS ENGINEERING APPROACH TO CONTINUOUS COMMAND AND CONTROL ON SECURITY-AWARE COMPUTER NETWORK

Nong Ye
Assistant Professor
Department of Mechanical Engineering
University of Illinois at Chicago

Abstract

It is of strategic importance to move command and control (C^2) from schedule-driven cyclic operations to situation-driven continuous operations so that C^2 can respond to new campaign objectives and situation updates in a timely fashion. The responsiveness of C^2 requires the integration of currently loosely connected C^2 operations into a coherent infrastructure and a continuous workflow that are supported by a security-aware computer network. The objective of my summer work at the Information Institute of the U. S. Air Force's Rome Laboratory is to investigate the feasibility of a process engineering approach to continuous command and control as well as network security management. The investigation focuses on the development of a process model for C^2 operations and a process model for network security management, based on the dynamic process modeling schema.

The process model for C^2 operations consists of a four-level abstraction hierarchy that supports the integration of C^2 operations across different levels (i.e., objectives, targets, tasks, actions, and events), different stages at each level (i.e., planning, scheduling, execution, monitoring, assessment, and replanning), and different functional areas at each level and in each stage (i.e., force application, force enhancement, and force support). The process model for network security management considers network intrusive behaviors as anomalies or deviations from the security specification and profiles of a computer network. This process model will provide a coherent infrastructure to integrate existing security management techniques. It will also guide future work concerning how existing techniques should be advanced, how different techniques should be integrated, what security policies should be specified, what network activities should be visualized for security management, what should constitute a rapid prototyping environment of security management system, and so on.

A PROCESS ENGINEERING APPROACH TO CONTINUOUS COMMAND AND CONTROL ON SECURITY-AWARE COMPUTER NETWORK

Nong Ye

1. Introduction

To improve the responsiveness to changing requirements and dynamic situations in air campaigns, command and control (C^2) must move from schedule-driven, cyclic operations to situation-driven continuous operations [1-6]. This requires the integration of C^2 operations across different levels (i.e., objectives, targets, tasks, actions, and events), different stages at each level (i.e., planning, scheduling, execution, monitoring, assessment, and replanning), and different functional areas at each level and in each stage (i.e., force application, force enhancement, and force support). The highly interleaved C^2 operations rely on a security-aware computer network that is protected against malicious attacks in information warfare [7-10].

Existing research and development efforts for C^2 operations and network security management lack a comprehensive and systematic infrastructure to integrate existing point solutions into a continuous and coherent process. The objective of this work is to develop a new perspective that models both an air campaign and a computer network as a dynamic process system. Techniques employed in this study are drawn from those used for the analysis and control of industrial systems. A process model allows the application of the well-developed process planning, control and diagnosis schema to C^2 operations and network security management. The process model also provides an integration infrastructure to incorporate existing point solutions into the process planning, control and diagnosis schema. This report presents a process model of air campaign for continuous command and control, as well as a process model of computer network system for security management. The implications of these process models in moving C^2 operations and network security management to highly responsive and integrated processes are discussed.

2. The Process Engineering Schema

The process engineering schema has traditionally been used for industrial systems (e.g., discrete-part or continuous process manufacturing systems) [11-12]. The planning, control and diagnosis of an industrial process system are conducted at four levels of abstraction: objective, conceptual, functional, and physical [13-14]. At the objective level, the goals of the system are stated. At the conceptual level, the goals of the system are transformed into time phased and sequenced tasks which are described by system states and state transitions. At the functional level, the conceptual model of the system is supported by actions that take place in a functional architecture of the system. The functional model of the system describes functional system components and their working interrelationships. At the physical level, the functional model of the system is implemented by events taking place in the physical anatomy of the system.

At each level of abstraction, local activities within individual components aggregate into global activities over the entire system. Hence, a process model has two dimensions: vertical involving levels of abstraction from physical to objective, and horizontal involving levels from components, to subsystems, and finally to the system.

3. A Process Model of Air Campaign

By considering an air campaign as a process system, the dynamics of the air campaign system can be modeled at four levels (objective, conceptual, functional and physical). Two representation tools (state transition and block diagrams) are suggested to define the process model at each level. A state transition diagram represents dependencies (e.g., sequencing) and relationships (e.g. temporal) of activities. A block diagram specifies the architecture of the air campaign system in which activities take place. Hence, a block diagram can be used to show a structured picture of the air campaign system and its state at a given time, while a state transition diagram can be employed to illustrate the evolution of the air campaign system throughout its life cycle. Time tags may be attached to a state in a state transition diagram to indicate the planned and actual times for achieving that state.

Entities in a block diagram may represent a system, its subsystems, and its components. An entity has its state and behavior. The normal or authorized state and behavior of an entity may be specified using a set of constraints. Constraints of an entity should be monitored against the state and behavior of an entity, and violations of constraints should be detected.

Entities can be implemented using objects in object-oriented programming. An object contains attributes and methods. For an object, the values of its attributes give its state, and its methods describe its behavior. The monitoring of constraints can be implemented using an internal routine within an object. This internal routine, called internal constraint sentinel, is shared by all methods of the object. Since a request for a method leads to a certain behavior of the object and causes the change of the object's state, the internal constraint sentinel routine is executed to check constraints against the state and behavior of the object whenever a method is requested. In the following sections, objects are used to represent entities in the process model.

3.1. The Objective Level of the Campaign Process Model

At the objective level, campaign objectives are defined to include a start state at the beginning of an air campaign, a goal state which is the end state of the air campaign, and intermediate states which reflect the accomplishment of different campaign objectives. For each campaign objective, multiple states may be defined for multiple targets that are identified to achieve the campaign objective. Relationships among air campaign objectives and targets are also specified.

Figure 1 shows a state transition diagram at the objective level of a process model that has been developed for an air campaign scenario. This scenario was used by the ARPI (ARPA/Rome Laboratory Planning Initiative) program [15]. The objectives of this air campaign are to rescue a delegation in a US Embassy under the control of a 'red' force, and to secure an airport also under the control of that force. A state transition diagram at the objective

level of the process model provides a more explicit representation of campaign objectives, targets, and especially their relationships than a simple list of campaign objectives and targets.

The objective level involves objects such as a theater of operations and targets in the theater. Those objects and their interactions form a block diagram at the objective level. Each object has attributes and methods of using the object. A state is defined by the values of attributes from all objects at a given time (state vector). Values of object attributes at the objective level are fused from values of object attributes at the conceptual level.

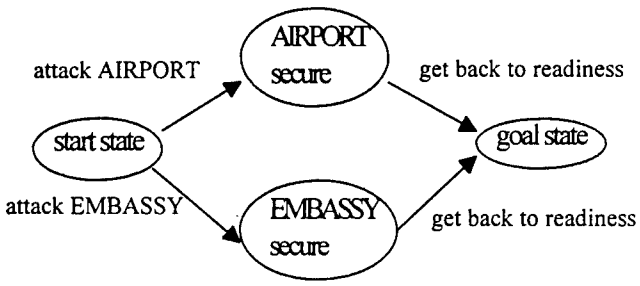


Figure 1. A state transition diagram at the objective level of the process model for the ARPI air campaign scenario.

3.2. The Conceptual Level of the Campaign Process Model

At the conceptual level, campaign objectives in terms of the consolidated state transition diagram are expanded into phased and sequenced states and *unified* joint force tasks for state transitions. Conceptual states reflect situations in the theater of operations at critical moments of an air campaign. Those moments identify different phases of the air campaign. All tasks leading to a state must be accomplished to realize that state. A unified joint force task is based on the composition of responsibilities from different functional areas (force application, force support and force enhancement. In other words, no distinction or division of responsibilities among different functional areas is made at the conceptual level. Hence, it is the joint force COA (courses of action) to achieve air campaign objectives that is defined at the conceptual level.

Figure 2 shows a state transition diagram at the conceptual level of the process model for the ARPI scenario. For example, a unified joint force task 'forces attack AIRPORT' changes the state of AIRPORT from controlled by a 'red' force to secured by a 'blue' force. This task requires a composition of actions from different functional areas (force application, force support and force enhancement). It should be noted that in a state transition diagram, the precedence of two states on a time scale is indicated not by the relative positions (e.g., left and right) of the nodes representing the states but by the direction of the arrow on the line connecting the nodes. It is possible that a loop may exist in a state transition diagram.

The conceptual level involves objects such as joint blue forces, joint red forces, and so on. Those objects and their relationships form a block diagram at the conceptual level. Each object has attributes and methods of using the object. A state can be described by a state vector which contains values of object attributes at a given time. Values of object attributes at the conceptual level are fused from values of object attributes at the functional level.

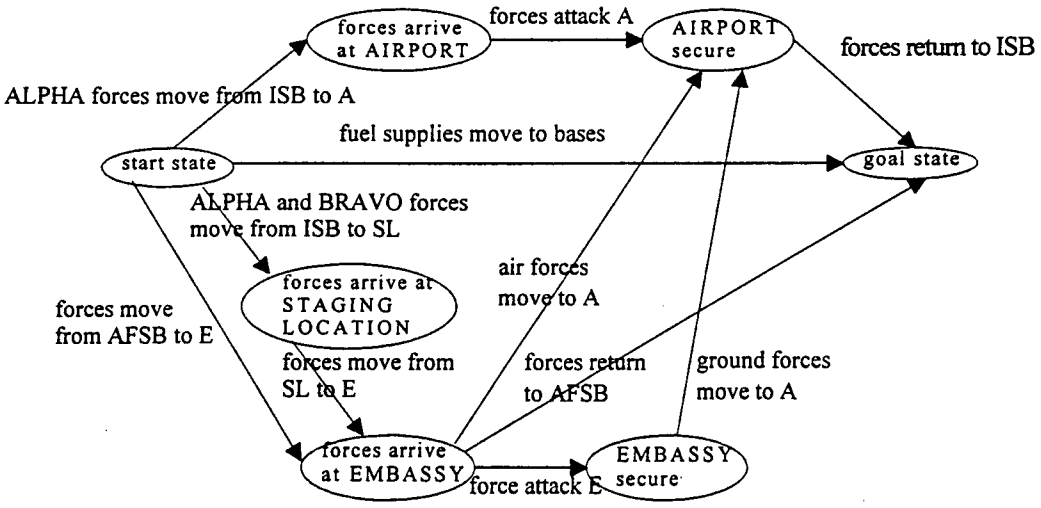


Figure 2. A state transition diagram at the conceptual level of the process model for the ARPI air campaign scenario.

3.3. The Functional Level of the Campaign Process Model

At the functional level, each unified task is decomposed into separate but coordinated actions that are executed by units in different functional areas (force application, force enhancement, and force support). Figure 3 shows a block diagram of the air campaign system at a given time for the functional level of the process model. The timing (e.g., actual and planned) of actions in the block diagram is maintained by objects at the functional level. A state transition diagram can also be used to explicitly represent dependencies of functional actions.

There are three types of objects at the functional level of the campaign process model: stationary objects, mobile objects, and path objects. Stationary objects are analogous to machines and devices in an industrial process system. They perform actions which then change their state. Examples of stationary objects in an air campaign system are bases, staging areas, ports, storage sites, supply stations, support headquarters, combat headquarters, etc. Actions may fall in the categories of mobilization, deployment, employment, sustainment, redeployment, etc. Mobile objects are analogous to electrical signal and data, mechanical motions, and flows of fluid that are transmitted among machines and devices in an industrial process system. Some examples of mobile objects in an air campaign system are lift and carrying assets (ships, trucks, trains, aircraft, etc.), materials (fuels, munitions, ammo, food, medical supplies, finance, containers, brake pads, personnel, cargo, palette, etc.), and force units (transportation units, supply units, maintenance units, combat units, etc.). Path objects are analogous to cables, wires, pipes and mechanical links in an industrial process system. Path objects provide channels for the movement of mobile objects among stationary objects. It should be noted that the type of object is determined based on the specific context of an air campaign system. For example, in one air campaign system a target may be a stationary object, but in another air campaign system a target is a mobile object.

t = 7
action O = 4 x CH47 arrive at AIRPORT
action P = 4 x UH607 arrive at AFSB

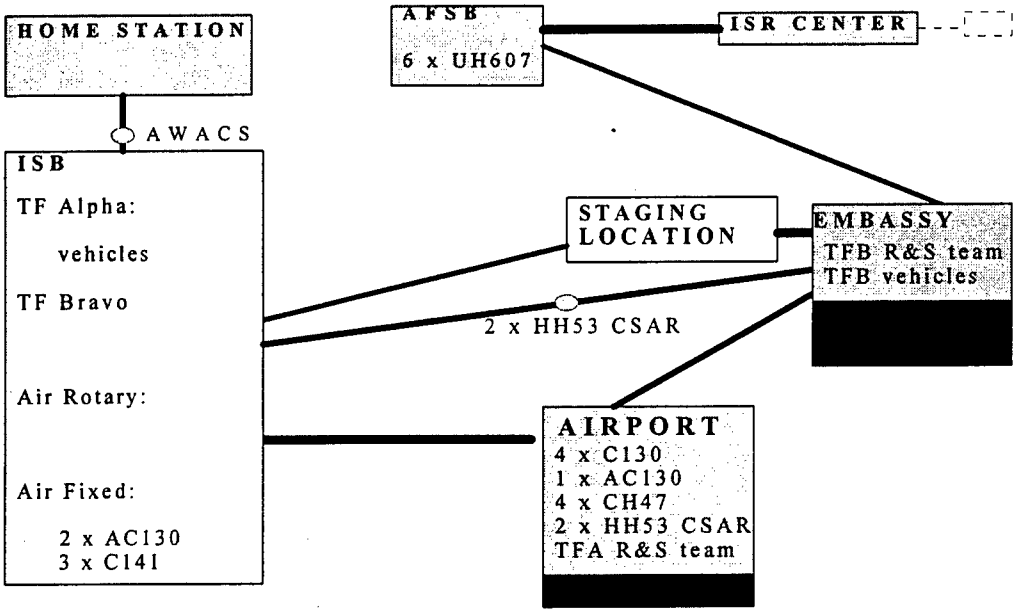


Figure 3. A block diagram at the functional level of the process model for the ARPI air campaign scenario.

Each object has attributes and methods of using that object. Values of object attributes at the functional level are fused from values of object attributes at the physical level. A stationary object AFSB (which is a base), a mobile object UH607, and a path object in the air campaign system for the ARPI scenario are illustrated below.

AFSB

Attributes

possession: (UH607, 6)
actual_deployment_time:
planned_deployment_time:
actual_return_time:
planned_return_time:
...

Methods

deploy: IF time = planned_deployment_time & UH607 is ready THEN send out 4 x UH607
ELSE set deployment-delay flag (a trigger for possible replanning).
return: IF time = planned_return_time THEN check posession.UH607 = 6;
IF input = 4 x UH607 THEN posession.UH607 = posession.UH607 + input.
report-available-forces (type_of_target): report available forces or weapons to assault the type_of_target.
approach: IF input = ISR signal of approach red force THEN prepare for counter air.

attack:
defend:
...

Internal Constraint Sentinel (executed when each of the methods is requested)
IF possession.UH607 < 3, THEN set risk flag;
...

UH607

Attributes
role: list of targets that it can assault and its effectiveness for each type of targets
origin: AFSB
destination: EMBASSY
current_position:
planned_depart_time:
actual_depart_time:
planned_arrival_time:
actual_arrival_time:

Methods

receive-order:	planned_depart_time = X; planned_arrival_time = X.

take-off:	origin = deploy.origin;
	destination = deploy.destination;
	actual_depart_time = X.

in-flight:	LOOP maintain current_position from ISR data; IF ISR data gives a HALT signal THEN set HALT flag and jump out of the loop; ELSE calculate transmission rate; project the estimated arrival time; IF the estimated arrival time > planned_arrival_time + threshold THEN set the delay flag; ENDLOOP when current_position = destination.
land:	actual_arrival_time = X.
report_role:	report types of target that it can assault and effectiveness

...

Internal Constraint Sentinel
...

PATH-1

Attributes
capacity:
time_of_availability:
list of mobile components and their current positions:
...

Methods

Establish:	time_of_availability = X.
Use:	add the name and position of mobile object to the list.

...

Internal Constraints Sentinel
...

A functional path corresponds to a physical route which may include river crossing, road following, etc. At the functional level, the physical route is not of interest. There is a link from a functional path to its corresponding physical route at the physical level. Hence, relative positions of objects in the functional model do not necessarily reflect physical (e.g., geographical) locations and configurations of objects in the theater of operations.

Each object has a set of constraints which determine its survivability requirements in the air campaign system, for example, vulnerability to attack which is analogous to vulnerability to failure and malfunction in an industrial process system. Different degrees of survivability requirements can be defined. When survivability requirements of an object are not satisfied to a certain degree, a flag of warning to that degree is raised. A warning flag demands for a maintenance effort for an industrial process system and a campaign assessment and possibly replanning effort in an air campaign system. The internal constraint sentinel routine, which is attached to each method of an object, is responsible for checking constraints against current values of object attributes.

The arrival of a mobile object at a stationary object acts as an input to the stationary object, and *calls* a method of action on the stationary object. Whether or not an action can be *triggered* depends on the input and the current state of the stationary object. The current state of a stationary object is composed of current values of the object's attributes.

At a given time, a mobile object is attached to either a stationary object or a path object. The transmission or movement rate of a mobile object can be estimated from attribute values of the object during its movement. When an estimated arrival time deviates significantly from the planned arrival time, a delay flag is raised by the mobile object. The effect of the delay propagates to other objects to which the delayed mobile object lead directly and indirectly. The propagation of deviation effect helps in assessing the impact of deviation throughout the air campaign system. A deviation of an actual situation from a planned situation can occur to a mobile object (e.g., a ship is sunk), a stationary object (e.g., 'red' force gains the control of a 'blue' force's base), or a path object (e.g., a bridge is damaged).

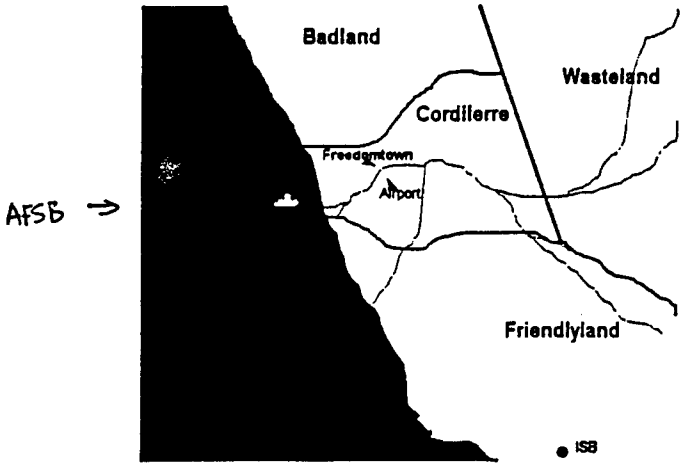
3.4. The Physical Level of the Campaign Process Model

At the physical level, functional objects in the air campaign system are broken down into physically identifiable objects which are represented in their raw physical forms, such as images of aircraft and targets, roads and bridges on a geographical map, terrain, weather, and so on. Hence, objects and their relationships are represented in their natural form at the physical level. Figure 4 shows a geographical map in the theater of operations for the ARPI scenario. A functional action is carried out as a series of physical events. A series of physical events for the functional action of air assault are shown in Figure 4. Each physical object has attributes and methods of using that object. Physical actions are based on methods of using physical objects in the air campaign system. Values of object attributes indicate the physical state in the air campaign system (theater of operations). Values of object attributes at the physical level are fused directly from ISR data which may be stored in databases of global awareness [16-17].

4. Implications of the Campaign Process Model to Continuous Command and Control

The process model of air campaign defines an air campaign system as a dynamic process system with levels of abstraction or granulation. Continuity is maintained for both representation and content of the process model from level to level, that is, from campaign objectives to phased and sequenced tasks of joint force, actions in

different functional areas, and finally to physical events. Since all stages (planning, scheduling, execution, monitoring, assessment and replanning) of air campaign operations in all functional areas are based on the same 'language' - the process model, continuity is also maintained across different stages of air campaign operations and across different functional areas at each level of abstraction. Continuity in the campaign process model provides the foundation for developing integrated C^2 operations. The campaign process model also helps in linking and integrating existing planning systems (e.g., TSA, SOCAP, ACPT) based on the coherent infrastructure of the campaign process model [18-19]. The process model of integrated C^2 operations can be implemented using an object-oriented methodology, from which common object schemas for command and control of air campaign can be derived. A state transition diagram or a block diagram presents a network of coordinated activities rather than a list of separate activities.



Events for Airfield Assault

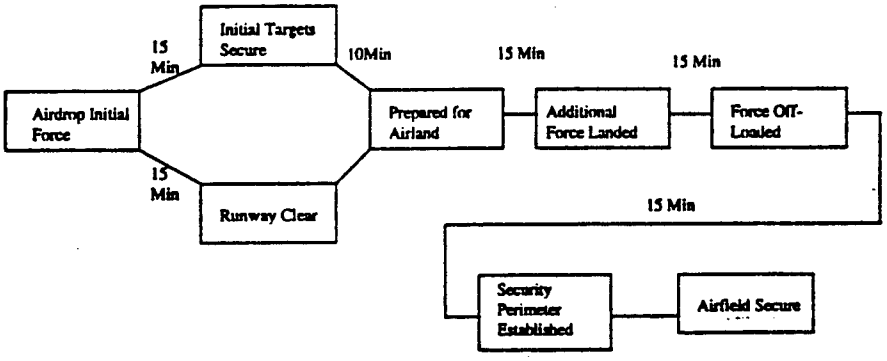


Figure 4. The geographic map of theater and a series of air assault events. for the ARPI air campaign scenario.

States in a state transition diagram provide potential decision points for campaign assessment. For example, the goal state of secured targets and force readiness at the objective level of the ARPI scenario can trigger a

campaign assessment at a planned time when this state should be accomplished. First of all, the current state vector at the objective level must be obtained from the values of object attributes at this level. To obtain the state vector at the objective level, an information need request is made by objects at the objective level to their supporting objects at the conceptual level for information fusion of object attributes from the conceptual level to the objective level. To meet the information need request from the objective level, those associated objects at the conceptual level must then obtain the values of their attributes from objects at the functional level. Hence, the information need request is propagated from objects at the objective level to associated objects at the conceptual level, then to associated objects at the functional level, and finally to associated objects at the physical level, through inter-level links among objects at different levels. Those objects at the physical level directly acquire their attribute values from the ISR databases. Information fusion is then carried out bottom up from the physical level, all the way to the objective level to satisfy the information need request which is initiated at the objective level. The current state vector at the objective level is then compared with the desired state vector at the planned time of accomplishing the goal state. Therefore, an information need request is propagated top down from the level that makes the request to lower levels, which in turn initiates bottom-up information fusion to meet the information need request for campaign assessment.

Information fusion may involve uncertainty, because an attribute value at a lower level may not reach a certain level of confidence at the time when this value is requested by a higher level. Hence, information fusion must cope with attribute values with uncertainty. Based on the process control schema, the bottom-up information fusion provides an observation of the current state of the air campaign system with measurement uncertainty due to measurement noises. To obtain a more accurate and reliable estimate of the current state of the air campaign system, system identification and estimation techniques such as Kalman filters can be utilized to estimate the current state of the air campaign system based on the consideration of both state observation and state prediction.

A prediction of the current system state relies on previous system states and a process model. The process model describes how an action on the system leads the system from a state at one time to a state at the next time. Hence, a state transition diagram provides a process model of air campaign. For example, the goal state at the objective level of the ARPI scenario can be predicted from the estimates of the two previous states 'AIRPORT secure' and 'EMBASSY secure' as well as the planned actions on the air campaign system leading from these two states to the goal state. Because of noises in the air campaign system, the actual goal state that results from the planned actions on the two previous states may deviate from the desired goal state. Hence, the state prediction also involves uncertainty due to system noises. Kalman filters estimate a system state based on a tradeoff between measurement noises and system noises.

Every state in a state transition diagram at each level can become a decision point to start an air campaign assessment at the planned time when that particular state should be accomplished. The estimate of a state is then used to estimate the next state. This enables continuous campaign assessment. The top-down propagation of information need and the bottom-up fusion of information can be carried out only at decision points for campaign assessment. In other words, the states in the state transition diagrams determine the rate of data sampling. Instead of updating the state of the air campaign system constantly all the time, information fusion for campaign assessment

can be carried out on an as-needed basis, such as at decision points as determined by the states in state transition diagrams.

5. A Process Model of Security-Aware Computer Network

The importance of computer network security has been well acknowledged. Intrusions to the secure operation of a computer network system may take many forms: external attacks, internal misuses, network-based attacks, host-based attacks, information gathering, access to protected services, denial of service, and so on.

5.1. Existing Techniques of Security Management

Existing techniques that protect a computer network system against intrusions fall in mainly two categories: prevention and detection [20]. Intrusion prevention techniques include the use of firewall, encryption, authentication, and so on [20]. A firewall builds a shield around a secure network system to intercept network traffic and reject requests for activities that are not authorized to take place. Either authorized or unauthorized activities can be specified based on source IP address, destination IP address, IP protocol, port, network adapter, and traffic direction. The firewall technique is similar to specification-based intrusion detection techniques, except that a firewall usually denies requests for unauthorized activities without informing the system administrator that a violation occurs. Both firewalls and specifications are mechanisms which set security policies for network systems.

Intrusion detection techniques detect suspicious intrusions to secure network operations while they are taking place, diagnose their cause, assess consequent damage, and respond to prevent further damage. Intrusion detection techniques generally fall within three categories: statistical-based, signature-based, or specification-based. Statistical-based techniques detect deviations from statistical profiles of normal activities which mostly are built on frequency distributions of those activities [21]. Signature-based techniques match system activities with well-defined signatures of known intrusions. Intrusion signatures may be characterized by strings [22] (e.g., names of privileged programs, strings in the header or data section of network packets), sequences of events [23], activity graphs [24], etc. Signature recognition can be performed using matching algorithms [23] and expert systems [25]. To capture sequences of events, Petri nets [23], state transition analysis [26], artificial neural networks [27] and time-based inductive generalization [28] have all been employed. Specification-based techniques detect deviations from a predicate logic based specification of normal behavior, for example, normal operations and their valid parameters of privileged programs.

Among the existing intrusion detection techniques, both specification-based techniques and signature-based techniques require a priori understanding of normal behavior or anomalies. For many individual components (e.g., routers, privileged programs) of network system, there are limited methods to request and use services of those components, and abnormal ways to attack those components have been widely observed. However, normal behavior and sophisticated intrusions at the system level may involve multiple components and their interactions. System-level normal behavior and anomalies are unpredictable and difficult to specify in advance.

Although frequency-based statistical profiling is able to capture the dynamic development of normal behavior, measures on which statistical profiles are built must be selected in advance. Different sets of measures may be needed to address different types of intrusions. A statistically effective differentiation of anomalies from normal behavior may become more difficult at the system level, because of a possibly larger gray area between statistical profiles of normal and abnormal behavior on collective measures. This further aggravates the problem of measure selection. Moreover, the frequency factor alone may not be sufficient to detect anomalies at the system level. Other factors such as the sequence and timing of events must also be taken into account.

Therefore, the existing intrusion detection techniques offer point solutions which are effective mainly at the component level for specific types, points and paths of intrusions. Those techniques are necessary elements of intrusion detection function, but are not sufficient for a comprehensive coverage of both component-level and system-level intrusions. The lack of an integration infrastructure to fuse results of point solutions into a more global awareness of intrusions adds another problem with selecting and using COTS (Commercial Off The Shelf) systems that are based on the existing intrusion detection techniques. System-level intrusion detection techniques are highly desirable to address more dynamic, sophisticated, unanticipated intrusions which are more likely to be used by organized attackers against network systems critical to the nation's security (e.g., those within the DOD, the Air Force, the Army and the Navy).

5.2. A Process Model of Security-Aware Network System

The process model is the basis to perform fault diagnosis for industrial systems. Faults are system anomalies or deviations from normal system behavior. If we consider intrusive behaviors as anomalies or deviations from the normal behavior of computer network system, it will allow us to apply well-developed models and techniques in the area of process control and diagnosis for detecting, diagnosing and responding to intrusions on computer network systems. Although the development of intrusion detection techniques is still in its infancy, decades of R&D in the area of industrial process control and diagnosis have produced fruitful outcome.

In collaboration with Rome Laboratory's Information Warfare Team, a process model of security-aware network system has been proposed to lay the foundation for detection, diagnosis and response of network intrusions. This process model will also serve as an integration infrastructure for incorporating and combining the existing intrusion detection techniques into a coherent architecture.

Fault diagnosis techniques vary from model-based diagnosis concerning normal system behavior, statistical-based diagnosis concerning normal behavior at critical points of system operation, causal graphs concerning causal relations of system abnormal behavior, to pattern recognition concerning known patterns of faults. Model-based diagnosis builds a model of normal system behavior (system model), detects deviations from normal system behavior, and diagnoses causes of detected deviations (symptoms of faults) by searching for system components whose faults account for all fault symptoms [29-32]. The search is based on the system model which describes normal behavior of each system component and normal relationships among system components. Statistical-based diagnosis makes statistical inference of system activities based on their statistical profiles, using

measures taken at certain points of system operation [33]. Causal graphs construct a structure of abnormal system behavior (e.g., fault tree or Bayesian network of fault effects appearing at various points of system operation) to describe cause-effect relationships of fault effects as they propagate through the system [34-42]. Fault diagnosis is performed by tracing from fault symptoms to fault causes along causal paths. Pattern recognition stores symptom-cause pairs, and diagnoses faults by matching symptom patterns [43].

In general, both model-based and statistical-based diagnosis are based on a model of normal behavior. They differ in that model-based diagnosis provides a global view of the system, whereas statistical-based diagnosis gives a local focus on specific points of system operation. Both causal graphs and pattern recognition concern abnormal behavior. However, causal graphs provide a global view of the system, whereas pattern recognition takes a local, isolated view. All these diagnostic techniques can be applied to fault diagnosis in any layer of the process model. For each layer, all these techniques are necessary to cover from normal to abnormal behavior and from global to local points of view.

Taking an analogy between the fault diagnosis techniques and the existing intrusion detection techniques, the gap between a comprehensive set of diagnostic techniques and the existing intrusion detection techniques becomes apparent. Statistical-based and signature-based detection techniques offer only local viewpoints of normal and abnormal behavior. Specification-based detection techniques (including firewalls) currently address only the component level, and still lack a global viewpoint. The analogy also indicates how the existing intrusion detection techniques fit in the process model of security-aware network system. Since the existing intrusion detection techniques offer local solutions, they can be embedded within individual components of network system to detect, diagnose, and respond to local attacks on those individual components.

Therefore, the process model provides an infrastructure to integrate the existing intrusion detection techniques. It also points out a need for system-level intrusion detection techniques.

Figure 5 shows a part of a process model for security-aware network system at the functional and conceptual levels, which has been developed in collaboration with Rome Laboratory's Information Warfare Team (RL/IWT). At the objective level, the goals of security-aware network system are confidentiality, availability, integrity, accountability, etc. The only object at this level is the system itself. At the conceptual level, objects are system states, and relationships among objects are presented in terms of state transitions. The functional level includes software objects, such as router, hosts, files, users, and so on. Relationships among functional objects may be data flows, control paths, etc. The physical level describes hardware resources as physical objects (e.g., computer CPU, memory boards, hard drive, and printer), physical actions operating on those objects, and physical interactions among those objects.

At the functional level of the process model (functional model) for security-aware network system, nodes represent functional components of network system, and links specify relationships among system components. There is a 'network' node which captures system-level activities. Security-related activities within a component are also illustrated through the ROOT USER object below.

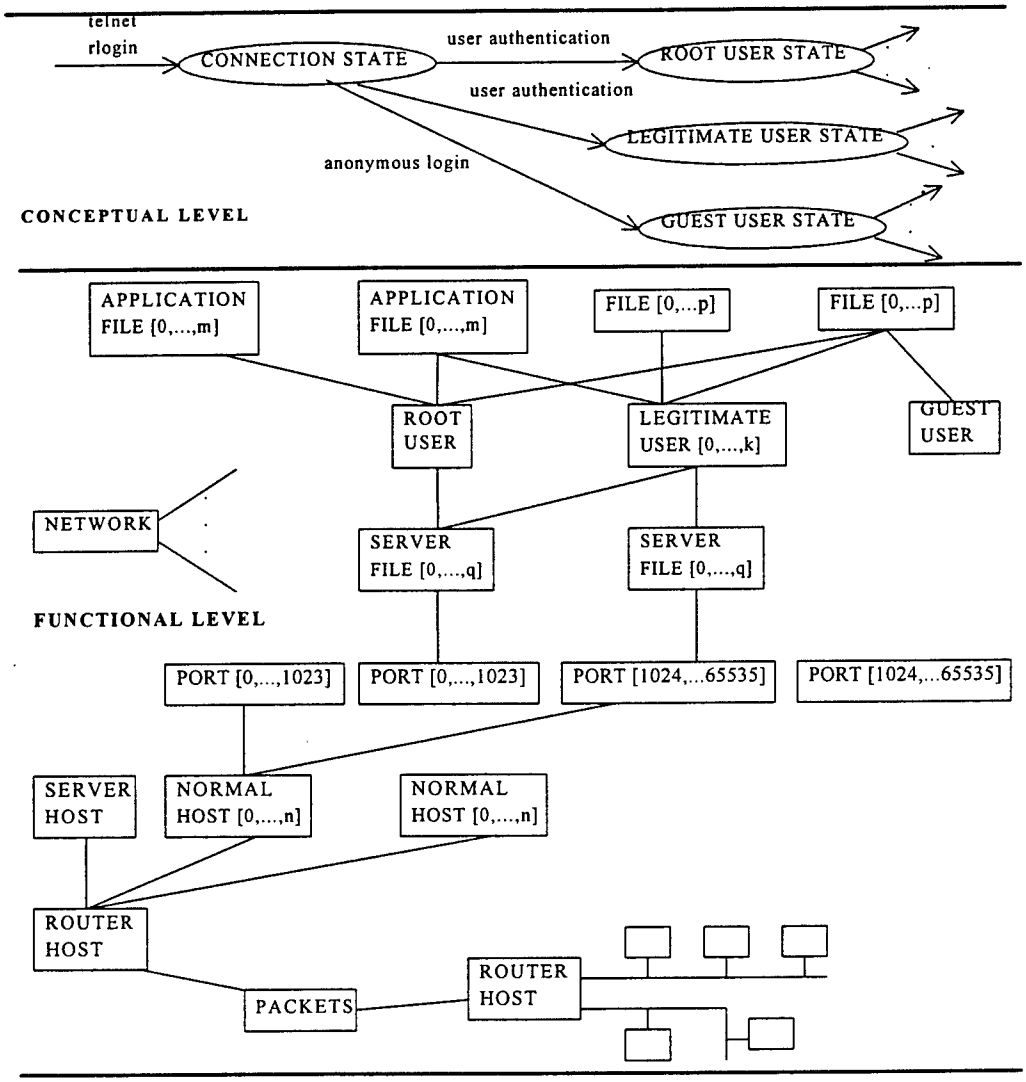


Figure 5. A part of a process model for security-aware computer network.

ROOT USER

Attributes

User id: root
Passwd: XXXXXXXX
Maximum number of active processes:
List of active processes:
History:
Deny list of hosts:
Allow list of hosts:
...

Methods

Rlogin(orig_Hname, requesting_user password, time, ...):
Telnet(orig_Hname, requesting_user, password, time, ...):
Rsh(orig_Hname, requesting_user, password, time, ...):
Ftp(orig_Hname, requesting_user, password, time, ...):
Finger (orig_Hname, ...):
...

Internal Constraint Sentinel

Specification of security policies (using rules, predicates, or other languages):

If user_name = 'root' & password = passswd & orig_Hname in Allow list & orig_Hname not in Deny list, I&W rating = 0.0 (allow);

If orig_Hname in Deny list, I&W rating = 1.0 (deny);

If more than 3 consecutive denies from the same host, I&W rating = 0.6 (alert);

...

Statistical profiling:

Signature recognition:

...

Data fusion:

Each component maintains data, allows methods to request services from the component, and has an internal constraint sentinel to detect, diagnose and respond to anomalies involving only the component itself. The internal constraint sentinel is executed whenever a method is requested. An intrusion detection function can be included for each local intrusion detection technique. Different intrusion detection techniques can be considered as different sensors for the same activity that yield different I&W (Indications and Warning of intrusion) ratings of the same activity with different levels of confidence. Those sensors can run in parallel or in sequence. A data fusion routine can be created to fuse the results from different sensors for an integrated I&W value of certain confidence associated with that component. A response action will be based on decision thresholds for the integrated I&W value and the confidence level. The I&W rating from an intrusion detection technique can be determined based on the degree of deviation from normal behavior or match with signatures. There is a well-developed body of sensor fusion techniques (e.g., Bayes' rule) to support the fusion of data from sensors which are different local intrusion detection techniques in this situation.

The implementation of the component-level intrusion detection function can be centralized at one place, or be distributed as agents within individual components. Data on system activities can be distributed to agents automatically or upon request. An object-oriented methodology can very well support the implementation.

The conceptual level captures system-level activities in more abstract terms using system states and state transitions. A system state is a characterized, collective measurement of system variables at a time.

If the component-level intrusion detection function offers the first line of defense against simple attacks, the system-level intrusion detection function provides the second line of defense against sophisticated attacks. A component-based I&W value may not be high enough to trigger a warning of a threat. System-level intrusion detection techniques, however, can fuse low component-level I&W values from many system components into more reliable, accurate, globally aware assessment of intrusion. The development of system-level intrusion detection techniques is imperative for reducing false alarms and relieving the system security analyst from dealing with too many alerts or messages triggered by low component-level I&W values. System-level intrusion detection techniques must deal with the dynamic nature of system behavior. First of all, system-level intrusions are inevitable but unpredictable because there are unlimited ways to organize system-level intrusions. This prevents the specification of system behavior in advance. Secondly, system behavior is dynamically changing.

6. Implications of the Network Process Model for Security Management

The potential benefits of a process model for security-aware network systems are listed below.

- Provide an integration infrastructure for the existing intrusion detection techniques.
- Clarify the limitation of the existing intrusion detection techniques and the future direction of research on intrusion detection.
- Help in determining what minimal set of data must be collected from a network system for maximally effective intrusion detection functions, based on the organization of security-related activity data in the process model of network system.
- Provide the guidance on how security-related system activities and results of the intrusion detection function should be visually presented to the human security analyst. Even if the intrusion detection function is not automated, the well-structured visualization of system activities in the process model will assist the human security analyst to look for right data and perform the intrusion detection function manually and effectively.
- Allow a systematic classification of intrusions based on types of entities involved as well as types and ranges (e.g., component-level versus system-level, functional layer versus conceptual layer) of anomalies.
- Guide the specification of security policies for computer network systems and the coordination of different intrusion detection techniques.
- Help in constructing a rapid prototyping environment by incorporating generalized classes of entities (e.g., user class generalized from the root user entity, legitimate user entities and guest user entities) in different layers and different levels of the process model into this environment.
- Facilitate the selection and application of COTS systems for the intrusion detection function of networks.
- Provide the directions of future R&D effort concerning, for example, what intrusion detection techniques should be developed, how different intrusion detection techniques should be integrated, what should be automated, what COTS systems should be considered, what should be specified, what should be visualized, what should constitute a rapid prototyping environment, and so on.

R&D effort based on the process model of security-aware computer network will lead to the flexibility, extendibility, dynamic adaptation and robustness (through distributed architecture) of the network security architecture and functionality.

Acknowledgment

This work is performed in collaboration with Rome Laboratory's Information Institute, Information Warfare Team, and JFACC program personnel. I would like to express my deep appreciation of gracious inputs, encouragement and support from Heather Dussault, Joseph Giordano and Jerry Dussault. My gratitude also goes to John Feldman, Gary Palmer, Charles Green, Mike Weir and Martin Sheppard of Information Warfare Team for their kind assistance and collaboration. Working with all of them has been one of my greatest experience with pleasure, excitement and productivity.

References

[1] JFACC Campaign Assessment Kick-Off Slides, TEKNOWLEDGE, ISX, PSR, and Logicon, June 17, 1997.

[2] Intelligence, Surveillance, and Reconnaissance (ISR) Planner Operational Concept Description (OCD), Logicon, 7 February 1997.

[3] JFACC online, http://jfacc6.dise.rl.af.mil.

[4] New World Vistas: Air and Space Power for the 21st Century, Summary Volume.

[5] Jon L. Boyes and Stephen J. Andriole (eds.), 1987, Principles of Command and Control. Washington, DC: AFCEA International Press.

[6] Stephen J. Andriole (ed.), 1990, Advanced Technology for Command and Control System Engineering. Fairfax, VA: AFCEA International Press.

[7] *Information Protection for the 21st Century*, AF Information Warfare Center, May 1997.

[8] *Security Architecture for the AITS Reference Architecture,* DARPA ISO Architecture Integrated Project Team, June 1997.

[9] *Rome Laboratory Defensive Information Warfare Program,* Presentation Slides by Joe Giordano, Rome Laboratory.

[10] *Future Intrusion Detection Methods,* Presentation Slides by Teresa Lunt, DARPA.

[11] Rasmussen, J. and Rouse, W. B., 1981, Human Detection and Diagnosis of System Failures (Plenum Press, New York).

[12] Ye, N. and Zhao, B. (1996), A hybrid intelligent system for fault diagnosis of advanced manufacturing system, International Journal of Production Research, 34(2), 555-576.

[13] Rasmussen, J., 1986, Information Processing and Human-Machine Interaction (North-Holland, New York).

[14] Ye, N. (1996), A hierarchy of system-oriented knowledge for diagnosis of manufacturing system faults, Information and System Engineering, 2(2), 79-103.

[15] Austin Tate (ed.), 1996, Advanced Planning Technology: Technological Achievements of the ARPA/Rome Laboratory Planning Initiative. Menlo Park, CA: AAAI Press.

[16] Richard T. Antony, 1995, Principle of Data Fusion Automation. Boston: Artech House.

[17] DMIF online, http://maco.dc.isx.com/iso/battle/dmif.html.

[18] ACPT (Air Campaign Planning Tool), http://isx.com/isxcorp/programs/acpt.html.

[19] SIPE-2 online, SOCAP online, http://www.ai.sri.com.

[20] White, G. B., Fisch, E. A., and Pooch, U. W. (1996). *Computer System and Network Security.* Boca Raton: CRC Press.

[21] *Safeguard Final Report: Detecting Unusual Program Behavior Using the NIDES Statistical Component,* SRI Project 2596.

[22] *Product Security Assessment of the NetRanger Intrusion Detection Management System,* Version 1.1., AF Information Warfare Center, February 1997.

[23] Kumar, S. (1995). *Classification and Detection of Computer Intrusions.* Ph.D. Dissertation, Purdue University.

[24] http://seclab.cs.ucdavis.edu.

[25] *Next-Generation Intrusion Detection Expert System (NIDES): A Summary*, SRI-CSL-95-07.

[26] Porras, P. A., and Kemmerer, R. A. (1992). Penetration state transition analysis - A rule-based intrusion detection approach. In *Eighth Annual Computer Security Applications Conference,* pp. 220-229. IEEE Computer Society Press, November 30 - December 4, 1992.

[27] Fox, K. L., Henning, R. R., Reed, J. H., and Simonian, R. (1990). A neural network approach towards intrusion detection. In *Proceedings of the 13th National Computer Security Conference,* pp. 125-134. Washington, DC, October 1990.

[28] Teng, H. S., Chen, K., and Lu, S. C. (1990). Security audit trail analysis using inductively generated predictive rules. In *Proceedings of the Sixth Conference on Artificial Intelligence Applications,* pp. 24-29, Piscataway, New Jersey, March 1990. IEEE.

[29] Crow, J., and Rushby, J. (1994). Model-based reconfiguration: diagnosis and recovery. NASA Contractor Report 4596.

[30] Reiter, R. (1987). A theory of diagnosis from first principles. Artificial Intelligence, 32(1), 57-95.

[31] Hamscher, W., Console, L., and de Kleer, J., 1992, Model-Based Diagnosis (Morgan Kaufmann Publishers, San Mateo, CA).

[32] Sangwine, S. J., 1989, Deductive fault diagnosis in digital circuits: A survey, IEEE Proceedings, 136(6), 496-505.

[33] Himmelblau, D. M. (1978). Fault Detection and Diagnosis in Chemical and Petrochemical Processes. Elservier: Amsterdam.

[34] Adamson, M. S. and Ronerge, P. R., 1991, The development of a deep knowledge diagnostic expert system using fault tree analysis information, The Canadian Journal of Chemical Engineering, 69, 76-80.

[35] Chang, S. J., DiCesare, F., and Goldbogen, G., 1991, Failure propagation trees for diagnosis in manufacturing systems, IEEE Transactions on Systems, Man, and Cybernetics, 21(4), 767-775.

[36] Kirsch, H., and Kroschel, K. (1994). Applying Bayesian networks to fault diagnosis. In Proceedings of the IEEE Conference on Control Applications, 895-900.

[37] Kokawa, M., Miyazaki, S., Shingai, S., 1983, Fault location using digraph and inverse direction search with application, Automatica, 19, 729.

[38] Naruo, N., Lehto, M., Salvendy, G., 1990, Development of a knowledge-based decision support system for diagnosing malfunctions of advanced production equipment, International Journal of Production Research, 28(12), 2259-2276.

[39] Nicholson, A. E., and Brady, J. M. (1994). Dynamic belief networks for discrete monitoring. IEEE Transactions on Systems, Man, and Cybernetics, 24(11), 1593-1610.

[40] Peng, Y. and Reggia, J. A., 1990, Adductive Inference Models for Diagnostic Problem Solving (Springer-Verlag, New York).